

Non-Oscillatory Finite Volume Methods for Conservation Laws on Unstructured Grids

A Thesis submitted for the degree of
Doctor of Philosophy
at the University of Leicester

by

Terhemem Aboiyar
Department of Mathematics,
University of Leicester,
United Kingdom.

September 2008

Abstract

Non-Oscillatory Finite Volume Methods for Conservation Laws on Unstructured Grids

by

Terhemen Aboiyar

This work focuses on the use of polyharmonic splines, a class of radial basis functions, in the reconstruction step of finite volume methods.

We first establish the theory of radial basis functions as a powerful tool for scattered data approximation. We thereafter provide existing and new results on the approximation order and numerical stability of local interpolation by polyharmonic splines. These results provide the tools needed in the design of the **R**unge **K**utta **W**eighted **E**ssentially **N**on-**O**scillatory (**RK-WENO**) method and the **A**rbitrary high order using high order **D**ERivatives-**W**ENO (**ADER-WENO**) method. In the RK-WENO method, a WENO reconstruction based on polyharmonic splines is coupled with Strong Stability Preserving (SSP) Runge-Kutta time stepping.

The polyharmonic spline WENO reconstruction is also used in the spatial discretisation of the ADER-WENO method. Here, the time discretisation is based on a Taylor expansion in time where the time derivatives are replaced by space derivatives using the Cauchy-Kowalewski procedure. The high order flux evaluation of the ADER-WENO method is achieved by solving generalized Riemann problems for the spatial derivatives across cell interfaces.

Adaptive formulations of the RK-WENO and ADER-WENO methods are used to solve advection problems on unstructured triangulations. An *a posteriori* error indicator is used to design the adaptation rules for the dynamic modification of the triangular mesh during the simulation. In addition, the flexibility of the stencil selection strategy for polyharmonic spline reconstruction is utilised in developing a WENO reconstruction method with stencil adaptivity.

Finally, order variation procedures are combined with mesh adaptation in order to handle regions of the computational domain where the solution is smooth in a different fashion from the vicinity of singularities and steep gradients with the goal of delivering accurate solutions with less computational effort and fewer degrees of freedom when compared to adaptive methods with fixed order of reconstruction.

To
my parents,
Aaron and Cecilia
and
in memory of
David Mser (1959-2008)

Acknowledgements

I would like to thank all the good people who made this thesis, in particular, and my stay in Leicester, in general, a success.

I must particularly thank my supervisor, Dr. Emmanuil Georgoulis. His enthusiasm, inspiration, patience, critical thinking and his understanding nature helped me to get through even at the times I was discouraged. Throughout my thesis-writing period, he provided encouragement, sound advice, good supervision and lots of useful ideas.

I would also like to thank my first supervisor, Prof. Armin Iske who is now at Hamburg in Germany for putting me on this project. I appreciate his help and input during the period of my research. Despite the fact we did not communicate very much towards the end of my programme, I deeply appreciate the opportunity he gave me to carry out research on both finite volume methods and radial basis functions.

I am grateful to all the staff of the mathematics department of the University of Leicester. I particularly wish to thank Prof. Jeremy Levesley, the head of department, for his help and advice throughout my stay in Leicester.

I would like to extend my appreciation to all the other research students I met in the department. I am especially grateful to Dr. Jonathan Crofts, Dr. Ed Hall, Richard Handel, Warwick Dumas, Nicckiy Bhyat, Vasile Stanciulescu, Jianxia Zhang, Abdul Samad, Fazli Subhan, Rochelle Randall, Mohammad Amjad and Taheer Khan for their support and friendship. I also wish to thank Dr. Rob Brownlee for the enlightening discussions I had with him on the theory of radial basis functions and Dr. Martin Käser for taking time to explain the WENO and ADER algorithms to me.

To my friends - Martin & Marion Akiga, Malcolm Orkar, David Adebayo, Aniete Isong, Emmanuel & Winifred Agogo, Mwuese Hon, Jude Bwire and many others - I would like to say a big thank you.

I deeply appreciate the love and kindness of the Pastor and members of the Amazing Grace Parish, Leicester. They have been a big family to me for four years. May the Lord bless you all richly.

I owe a debt of gratitude to Prof. Peter Onumanyi, Prof. Richard Kimbir and Prof. Stephen Onah for the impact they have had on my mathematical career and for all the encouragement they have given me through the years.

I wish to express my deep appreciation to my parents, Aaron and Cecilia Aboiyar, my brothers, my sister and my extended family for their support, encouragement and prayers during this programme. I am forever indebted to Patience Iortim for her understanding, endless patience and encouragement when it was most required.

My profound gratitude goes to the Commonwealth Scholarship Commission in the United Kingdom for funding my studies in the UK. It was really an opportunity of a lifetime.

Finally, I wish to thank the University of Agriculture, Makurdi, Nigeria for granting me leave to undertake this programme.

Terhemem Aboiyar

Leicester, UK

September 2008

Contents

	i
Acknowledgements	iii
1 Introduction	1
1.1 Fundamentals	2
1.1.1 Derivation and basic concepts	2
1.1.2 Conservation laws in one space dimension	4
1.1.3 Multidimensional conservation laws	5
1.2 Numerical Methods	7
1.2.1 A survey of finite volume methods	8
1.3 Objectives, Outline and Main Results	11
1.3.1 Objectives and motivation	11
1.3.2 Main results	12
1.3.3 Outline of the thesis	14
2 Radial Basis Functions	16
2.1 Radial Basis Function Interpolation	16
2.1.1 The interpolation problem	17
2.1.2 Solving the interpolation problem	18
2.1.3 Characterization of conditionally positive definite functions	20
2.1.4 Lagrange form of the interpolant	21
2.1.5 The optimality of RBF interpolation	22
2.1.6 Numerical stability	22
2.1.7 Polyharmonic splines	23
2.2 Generalized Interpolation	24
2.2.1 Error estimates	26
2.3 Generalized Local Interpolation by Polyharmonic Splines	31
2.3.1 Local approximation order and numerical stability	33
2.3.2 Derivatives of polyharmonic splines	37

3	The RK-WENO Method	45
3.1	An Introduction to the Finite Volume Method	46
3.1.1	Numerical fluxes	47
3.2	Finite Volume Formulation on Triangular Meshes	48
3.2.1	Conforming triangulations	48
3.2.2	Semi-discrete formulation on triangular meshes	49
3.3	Time Discretisation	50
3.3.1	The CFL condition	51
3.3.2	Optimal SSP Runge-Kutta methods	51
3.4	Reconstruction of Polyharmonic Splines from Cell Averages	53
3.4.1	Reconstruction by radial basis functions	53
3.4.2	Reconstruction by polyharmonic splines	54
3.5	Local Lagrange Reconstruction by Polyharmonic Splines	55
3.6	Stencil Selection	57
3.6.1	Enhanced flexibility in stencil selection for polyharmonic splines	61
3.7	The WENO Reconstruction	61
3.7.1	The oscillation indicator and weights	63
3.8	Numerical Experiments	64
3.8.1	Numerical implementation	64
3.8.2	Linear advection	64
3.8.3	Burgers' equation	68
3.9	Treatment of Boundary Conditions	70
3.10	Doswell's Frontogenesis	72
4	The ADER-WENO method	79
4.1	The ADER-WENO Finite Volume Scheme	80
4.1.1	The Generalized Riemann Problem at the cell interface	82
4.1.2	Flux evaluation	83
4.1.3	Computing the derivatives	86
4.1.4	Algorithm for the ADER-WENO method	89
4.2	Numerical Examples	90
4.3	Smolarkiewicz's Deformational Flow Test	92
4.4	Some Comments on the RK-WENO and ADER-WENO Schemes	93
5	Adaptivity	97
5.1	The Error Indicator	98
5.2	Stencil Adaptivity	99
5.2.1	Example : Linear advection	100
5.2.2	Example : Burgers' equation	101

5.3	Mesh Adaptivity	103
5.3.1	Initial mesh adaptation	104
5.3.2	Conservative interpolation	105
5.3.3	Algorithm	106
5.4	Numerical Examples on Mesh Adaptivity	107
5.4.1	Burgers' equation	107
5.4.2	Time dependent velocity fields: The swirling deformation problem	112
5.4.3	Two-phase flow in porous media	114
5.5	Mesh & Stencil Adaptivity	124
6	Mesh & Order Adaptivity	126
6.1	The Refinement Strategy	128
6.2	Estimation of Smoothness	129
6.3	The Algorithm	129
6.4	Numerical Examples	131
6.4.1	Example 1: Linear advection	132
6.4.2	Example 2: Kinematic wave equation	134
6.4.3	Example 3: Burgers' equation	136
7	Conclusion and Outlook	139
7.1	Conclusion	139
7.2	Outlook	141
A	Notations and Abbreviations	143
A.1	Notations	143
A.2	Abbreviations	144
B	Newest Vertex Bisection	145
B.1	Bisection of Triangles	145
B.2	Coarsening Algorithm	148
C	Modelling Two-Phase Flow in Porous Media	149
C.1	Introduction	149
C.2	The Governing Equations	149
C.3	The Simplified Mathematical Model	151
	Bibliography	153

List of Figures

1.1	Some finite volume methods for the linear advection equation showing oscillations near discontinuities at time $t = 2$ and $N = 160$ gridpoints. The bold line represents the exact solution while the dotted line represents the finite volume approximation.	9
1.2	High resolution finite volume methods with slope limiters for the linear advection equation at time $t = 2$ and $N = 160$ grid points. The bold line represents the exact solution while the dotted line represents the finite volume approximation.	9
3.1	Two neighbourhoods of the triangle T (marked red).	58
3.2	The three forward sectors of triangle T	58
3.3	The three backward sectors of triangle T	59
3.4	Three centred stencils of size five for the triangle T (marked red).	60
3.5	Three forward stencils of size five for the triangle T (marked red).	60
3.6	Three backward stencils of size five for the triangle T (marked red).	60
3.7	(a) Velocity field and (b) initial condition for linear advection equation.	65
3.8	Sequence of four meshes with their mesh sizes.	66
3.9	Solution of (3.35), (3.36) at times $t = 0, 0.25, 0.5,$ and 1 using thin plate spline reconstruction.	67
3.10	Solution of (3.37), (3.39) on stencils of size four and seven at time $t = 1.49514$ using the RK-WENO method with thin plate spline reconstruction.	69
3.11	(a) Velocity field and (b) steady state solution for (3.40) - (3.42).	71
3.12	(a) Velocity field and (b) initial condition for Doswell's frontogenesis.	73
3.13	Specially created computational mesh for Doswell's frontogenesis.	74
3.14	Two meshes used for the solution of Doswell's frontogenesis.	74
3.15	Solution of Doswell's frontogenesis problem at (a) $t = 1$; (b) $t = 2$; (c) $t = 3$; (d) $t = 4$. Results obtained by RK-WENO method on the specially created mesh and with seven stencils of size seven.	75
3.16	Cuts along the x_2 -axis at time $t = 4$ for different solutions	76

3.17	Solution of Doswell's frontogenesis problem by RK-WENO method on specially created mesh with stencils of size four at $t = 4$	76
3.18	Solution of Doswell's frontogenesis at $t = 4$ on meshes with widths (a) $h = 1/40$ and (b) $h = 1/80$	77
4.1	Flux evaluation across the edge of a triangle from t^n to t^{n+1} using two Gaussian integration points in space and time.	81
4.2	The GRP along the outward pointing unit vector with polyharmonic spline reconstructions $s_L(x)$ and $s_R(x)$ approximating the solution u . . .	83
4.3	(a) Velocity field and (b) initial condition for Smolarkiewicz's deformational test.	92
4.4	Solution of Smolarkiewicz's deformational test using ADER-WENO method at six different times (a) $t = 3T/200$, (b) $t = 3T/100$, (c) $t = 9T/200$, (d) $t = 3T/50$, (e) $t = 4T/50$ and (f) $t = T/10$ where $T = 2637.6$	94
4.5	Cuts along the x_2 -axis for Smolarkiewicz's deformational flow.	95
5.1	A triangle T (marked red) with its Moore neighbours (marked green). . .	98
5.2	(a) Stencil size distribution and (b) number of cells with particular stencil size.	102
5.3	(a) A coarse base mesh and (b) the same mesh adapted to the initial condition of the swirling deformation flow problem with 5 refinement cycles.	104
5.4	Solution of Burgers' equation at times (a) $t = 0$, (b) $t = 0.4$, (c) $t = 0.8$, and (d) $t = 1.2$ using the RK-WENO method.	108
5.5	Adapted mesh for the solution of Burgers' equation at times (a) $t = 0$, (b) $t = 0.4$, (c) $t = 0.8$, and (d) $t = 1.2$ using the RK-WENO method.	109
5.6	Solution of Burgers' equation at times (a) $t = 0$, (b) $t = 0.4$, (c) $t = 0.8$, and (d) $t = 1.2$ using the ADER-WENO method.	110
5.7	Adapted mesh for the solution of Burgers' equation at times (a) $t = 0$, (b) $t = 0.4$, (c) $t = 0.8$, and (d) $t = 1.2$ using the ADER-WENO method. . .	111
5.8	(a) Number of cells for different WENO reconstructions and (b) the number of cells used in the RK-WENO and ADER-WENO methods with thin plate spline reconstruction.	112
5.9	(a) The 3D plot and (b) the filled contour plot for initial condition for the swirling deformation problem.	114
5.10	(a) Velocity field for time $0 \leq t \leq T/2$ and (b) for time $T/2 < t \leq T$ for the swirling deformation problem.	115
5.11	Adapted mesh for the solution of the swirling deformation problem at times (a) $t = 0.375$, (b) $t = 0.75$, (c) $t = 1.125$, and (d) $t = 1.5$	116

5.12	Filled contour plots of the solution of the swirling deformation problem at times (a) $t = 0.375$, (b) $t = 0.75$, (c) $t = 1.125$, and (d) $t = 1.5$	117
5.13	3D plots of the solution of the swirling deformation problem at times (a) $t = 0.375$, (b) $t = 0.75$, (c) $t = 1.125$, and (d) $t = 1.5$	118
5.14	Number of cells during the simulation of the swirling deformation problem.	118
5.15	(a) Velocity field and (b) filled contours of the pressure field for the five spot problem.	119
5.16	Adapted mesh for the solution of the five spot problem by the ADER-WENO method at six different times (a) $t = t_0$, (b) $t = t_{490}$, (c) $t = t_{980}$, (d) $t = t_{1470}$, (e) $t = t_{1960}$, and (f) $t = t_{2450}$	121
5.17	Colour plots showing the water saturation u of the five spot problem at six different times (a) $t = t_0$, (b) $t = t_{490}$, (c) $t = t_{980}$, (d) $t = t_{1470}$, (e) $t = t_{1960}$, and (f) $t = t_{2450}$	122
5.18	Number of cells during the simulation of the five spot problem.	123
5.19	(a) Number of cells for mesh adaptivity and (b) number of cells and stencil size distribution for mesh & stencil adaptivity for the linear advection equation.	124
5.20	(a) Number of cells for mesh adaptivity and (b) number of cells and stencil size distribution for mesh & stencil adaptivity for Burgers' equation.	125
6.1	Mesh & order adaptivity mesh for linear advection problem. The red, green and blue cells are for reconstruction with $\phi_1(r)$, $\phi_{2,2}(r)$, $\phi_{2,3}(r)$ respectively.	133
6.2	Mesh & Order adaptivity mesh for linear advection problem. The red, green and blue cells are for reconstruction with $\phi_1(r)$, $\phi_{2,2}(r)$, $\phi_{2,3}(r)$ respectively.	133
6.3	Mesh & Order adaptive mesh for kinematic wave equation. The red, green and blue cells are for reconstruction with $\phi_1(r)$, $\phi_{2,2}(r)$, $\phi_{2,3}(r)$ respectively.	135
6.4	Mesh & Order adaptive mesh for kinematic wave equation. The red, green and blue cells are for reconstruction with $\phi_1(r)$, $\phi_{2,2}(r)$, $\phi_{2,3}(r)$ respectively.	135
6.5	(a) Initial condition and (b) solution at $t = 0.1$ of the kinematic wave equation.	136
6.6	Mesh & Order adaptive mesh for Burgers' equation. The red, green and blue cells are for reconstruction with $\phi_1(r)$, $\phi_{2,2}(r)$, $\phi_{2,3}(r)$ respectively.	137
6.7	Mesh & Order adaptive mesh for Burgers' equation. The red, green and blue cells are for reconstruction with $\phi_1(r)$, $\phi_{2,2}(r)$, $\phi_{2,3}(r)$ respectively.	137

B.1 Newest vertex bisection. (a) Labeled triangle T , (b) children of T , (c) marking for conformity, (d) bisection for conformity, (e) left child bisection and (f) right child bisection. 146

List of Tables

2.1	Radial basis functions (RBFs) and their orders.	17
3.1	Linear advection. Results by RK-WENO method using $\phi_1(r) = r$	65
3.2	Linear advection. Results by RK-WENO method using $\phi_{2,2}(r) = r^2 \log(r)$	65
3.3	Linear advection. Results by RK-WENO method using $\phi_{2,3}(r) = r^4 \log(r)$	65
3.4	Burgers' equation. Results by RK-WENO method using $\phi_1(r) = r$	68
3.5	Burgers' equation. Results by RK-WENO method using $\phi_{2,2}(r) = r^2 \log(r)$	68
3.6	Burgers' equation. Results by RK-WENO method using $\phi_{2,3}(r) = r^4 \log(r)$	69
3.7	Solution of (3.40) - (3.42) by the RK-WENO method using thin plate spline reconstruction.	72
4.1	Linear advection. Results by ADER-WENO method using $\phi_{2,2}(r) = r^2 \log(r)$ reconstruction.	91
4.2	Linear advection. Results by ADER-WENO method using $\phi_{2,3}(r) = r^4 \log(r)$ reconstruction.	91
4.3	Burgers' equation. Results by ADER-WENO method using $\phi_{2,2}(r) = r^2 \log(r)$ reconstruction.	91
4.4	Burgers' equation. Results by ADER-WENO method using $\phi_{2,3}(r) = r^4 \log(r)$ reconstruction.	91
5.1	Comparing the solution of the linear advection equation on (a) fixed stencil sizes and on (b) adapted stencils.	100
5.2	Comparing the solution of Burgers' equation on (a) fixed stencil sizes and on (b) adapted stencils.	101
5.3	Comparing mesh adaptivity and mesh & stencil adaptivity for the linear advection equation.	124
5.4	Comparing mesh adaptivity and mesh & stencil adaptivity for Burgers' equation.	125
6.1	Error comparison for mesh-only (m-o) and mesh & order (m & o) adaptivity for the linear advection equation. Two refinement cycles for each time step.	132

6.2	Error comparison for mesh-only (m-o) and mesh & order (m & o) adaptivity for the linear advection equation. Three refinement cycles for each time step.	132
6.3	Error comparison for mesh-only (m-o) and mesh & order (m & o) adaptivity for the kinematic wave equation. Two refinement cycles for each time step.	134
6.4	Error comparison for mesh-only (m-o) and mesh & order (m & o) adaptivity for the kinematic wave equation. Three refinement cycles for each time step.	134
6.5	Error comparison for mesh-only (m-o) and mesh & order (m & o) adaptivity for the Burgers' equation. Two refinement cycles for each time step.	136
6.6	Error comparison for mesh-only (m-o) and mesh & order (m & o) adaptivity for the Burgers' equation. Three refinement cycles for each time step.	136

Chapter 1

Introduction

A broad spectrum of problems in science and engineering are modelled with time-dependent hyperbolic conservation laws. Some of these problems arise in the fields of fluid mechanics, meteorology, reservoir modelling, compressible gas dynamics, traffic flow and in numerous biological processes. Examples of hyperbolic conservation laws include the Euler equations for gas dynamics, the shallow water equations, the equations of magnetohydrodynamics, the linear advection equation, the inviscid Burgers' equation and the Buckley-Leverett equation for flow in porous media. There are certain special properties and mathematical difficulties linked with these equations such as the formation of discontinuous solutions (shock waves, contact discontinuities, etc.) and nonuniqueness of solutions. These features need to be treated with care whenever numerical methods for hyperbolic conservation laws are developed. Fortunately, the rich mathematical structure of these equations can be used as a tool for developing efficient numerical methods. Moreover, when developing numerical methods for this class of problems care must be taken so that the presence of a discontinuity in the numerical solution does not induce spurious oscillations that affect the overall quality of the approximation. The methods also have to be sufficiently accurate near the discontinuity in order to clearly reflect the nature of the exact solution. To this end, in the past few decades, a large class of high order and high-resolution methods have been developed to handle the discontinuous solutions that are typical of hyperbolic conservation laws, while providing high order convergence rates. There has also been a growing interest in the development of genuinely multidimensional methods that are capable of capturing the geometrically complex interaction of linear and nonlinear waves.

In this thesis, we utilise radial basis functions in the reconstruction step of the spatial discretization of finite volume methods for the numerical solution of conservation laws. This is done within the framework of the Weighted Essentially Non-Oscillatory (**WENO**) reconstruction method. We will first combine this novel WENO reconstruction with Runge-Kutta time stepping where resulting numerical scheme is known as the Runge-Kutta WENO (**RK-WENO**) method. We will thereafter combine the

WENO reconstruction with the **ADER** (Arbitrary high order method using high order **DER**ivatives) time discretisation and flux evaluation strategy yielding the **ADER-WENO** method. Furthermore, we will implement adaptive algorithms in different contexts for the RK-WENO and ADER-WENO methods as a strategy for improving accuracy and reducing computational cost for problems with strong variations in their solutions.

Usually, when the WENO reconstruction method is combined with Runge-Kutta time stepping in the literature, the numerical scheme is also referred to as the WENO method while when the WENO reconstruction method is combined with the ADER time discretisation, the resulting method is simply referred to as the ADER scheme. However, we refer to these methods in this thesis as the RK-WENO and ADER-WENO methods to create a clear distinction between the two different settings within which the WENO reconstruction is used.

A list of the important notations and abbreviations used in this thesis can be found in Appendix A.

1.1 Fundamentals

1.1.1 Derivation and basic concepts

We consider a quantity Q in a region Ω in \mathbb{R}^d , $d = 1, 2, 3$, and we suppose the amount of Q contained in Ω need not be constant but can change with time. However, we assume that the amount of change is due only to the flow of Q across the boundary of Ω . These assumptions then provide a basis for the derivation of a conservation equation. Let the density of Q at position $\mathbf{x} \in \mathbb{R}^d$ and at time t be a scalar valued function denoted $u(t, \mathbf{x})$ and let $F = F(u(t, \mathbf{x}))$ be the flux field for Q . Then at time t , the amount of Q in an arbitrary ball \mathcal{B} in Ω is given by

$$\int_{\mathcal{B}} u(t, \mathbf{x}) d\mathbf{x}.$$

Similarly, the outflow through the boundary of the ball during a time interval $(t, t + \Delta t)$ is given by

$$\int_t^{t+\Delta t} \int_{\partial\mathcal{B}} F(u(t, \mathbf{x})) \cdot \mathbf{n} ds dt,$$

where \mathbf{n} denotes the outward unit normal to $\partial\mathcal{B}$, the boundary of the surface of the ball. The conservation law equation can then be formulated as:

$$\int_{\mathcal{B}} u(t + \Delta t, \mathbf{x}) = \int_{\mathcal{B}} u(t, \mathbf{x}) - \int_t^{t+\Delta t} \int_{\partial\mathcal{B}} F(u(t, \mathbf{x})) \cdot \mathbf{n} ds dt. \quad (1.1)$$

Using the fundamental theorem of calculus and divergence theorem we can write (1.1) as

$$\int_{\mathcal{B}} [u_t(t, \mathbf{x}) + \nabla \cdot F(u(t, \mathbf{x}))] d\mathbf{x} = 0. \quad (1.2)$$

Now since (1.2) must hold for every ball \mathcal{B} contained in Ω , and every time interval $(t, t + \Delta t)$, it follows that the differential form of the *conservation law* can be expressed as a Cauchy problem as follows:

$$\frac{\partial u}{\partial t} + \nabla \cdot F(u) = 0 \quad \text{in } \mathbb{R}^+ \times \mathbb{R}^d, \quad (1.3)$$

$$u(0, \mathbf{x}) = u_0(\mathbf{x}) \quad \text{in } \mathbb{R}^d. \quad (1.4)$$

In (1.3) - (1.4), $u(t, \mathbf{x})$ is the solution, $F(u) = (f_1(u), \dots, f_d(u))^T$ the flux function and $u_0(\mathbf{x})$ the initial condition. The function u is called the *classical solution* of the scalar problem if $u \in C^1(\mathbb{R}^+ \times \mathbb{R}^d)$ satisfies (1.3) - (1.4) pointwise. A well known property of nonlinear conservation laws is that the gradient of u may blow up in finite time even if the initial data u_0 is smooth. Thus, after a certain time t_b classical solutions for (1.3) - (1.4) may not exist, in general. This motivates the need for defining *weak solutions* which allow us to generalize the notion of solutions of conservation laws.

Definition 1.1 (Weak Solution) *Let $u_0 \in L^\infty(\mathbb{R}^d)$. Then u is called a weak solution of (1.3) - (1.4) if and only if $u \in L^\infty(\mathbb{R}^+ \times \mathbb{R}^d)$ and*

$$\int_{\mathbb{R}^d} \int_{\mathbb{R}^+} \left(u \frac{\partial \varphi}{\partial t} + F(u) \cdot \nabla \varphi \right) dt d\mathbf{x} + \int_{\mathbb{R}^d} \varphi(0, \mathbf{x}) u_0 d\mathbf{x} = 0 \quad (1.5)$$

for all $\varphi \in C_0^\infty([0, \infty) \times \mathbb{R}^d)$.

It is evident that (1.5) implies that u satisfies (1.3) - (1.4) in the sense of distributions. Thus (1.5) and (1.3) - (1.4) have meaning in the distributional sense even when the function u is discontinuous. A weak solution that lies in $C^1([0, \infty) \times \mathbb{R}^d)$ satisfies (1.3)-(1.4), i.e. it is also a classical solution. Furthermore, it is well known that weak solutions are often not uniquely defined [114]. To this end, a physically relevant weak solution can be selected from the collection of all possible solutions to a conservation law by using an additional constraint known as an *entropy condition*. The entropy condition takes into account the fact that physical processes are dissipative and that (1.3) models a physical process in the limit as the dissipation tends to zero [114]. This solution, the so-called *entropy solution* satisfies

$$\int_0^T \int_{\mathbb{R}^d} |u - c| \varphi_t + \sum_{i=1}^d \text{sign}(u - c) (f_i(u) - f_i(c)) \varphi_{x_i} d\mathbf{x} dt \geq 0 \quad (1.6)$$

for all non-negative test functions $\varphi \in C_0^\infty([0, T] \times \mathbb{R}^d)$ and all $c \in \mathbb{R}$. Condition (1.6) is also known as the *Kruzkov entropy condition*.

The following lemma summarizes some basic properties of solutions to (1.3) and (1.5).

Lemma 1.2 (Crandall and Majda [29]) *For every choice of initial data $u_0 \in L^\infty(\mathbb{R}^d) \cap L^1(\mathbb{R}^d)$, there exists a unique entropy solution $u \in C([0, \infty) : L^1(\mathbb{R}^d))$ of (1.3) with $u(0, \mathbf{x}) = u_0(\mathbf{x})$. Denoting this solution by $E(t)u_0$, we have:*

1. $\|E(t)u_0 - E(t)v_0\|_{L^1(\mathbb{R}^d)} \leq \|u_0 - v_0\|_{L^1(\mathbb{R}^d)}$,
2. $u_0 \leq v_0$ a.e. implies $E(t)u_0 \leq E(t)v_0$ a.e.,
3. $u_0 \in [a, b]$ a.e. implies $E(t)u_0 \in [a, b]$ a.e.,
4. If $u_0 \in BV(\mathbb{R}^d)$, $t \rightarrow E(t)u_0$ is Lipschitz continuous into $L^1(\mathbb{R}^d)$ and $\|E(t)u_0\|_{BV(\mathbb{R}^d)} \leq \|u_0\|_{BV(\mathbb{R}^d)}$.

where a.e. stands for almost everywhere.

We note that there are situations where the change in the density u of Q is also as a result of gains due to internal sources and sinks inside Ω which we denote by $S \equiv S(u(t, \mathbf{x}))$. This leads us to the equation

$$\frac{\partial u}{\partial t} + \nabla \cdot F(u) = S(u). \quad (1.7)$$

The equation (1.7) is called a *balance law* rather than a conservation law.

1.1.2 Conservation laws in one space dimension

Several properties of conservation laws can be clearly understood from the one dimensional conservation law

$$u_t + f(u)_x = 0. \quad (1.8)$$

The simplest example is the linear advection equation

$$u_t + \alpha u_x = 0, \quad (1.9)$$

where the Cauchy problem can be defined by this equation on $-\infty \leq x \leq \infty$, $t \geq 0$ together with the initial condition

$$u(0, x) = u_0(x).$$

The solution of this problem is $u(t, x) = u_0(x - \alpha t)$ and it is simply the initial data propagated unchanged with velocity α . The solution $u(t, x)$ is constant along the characteristics of the equation.

A famous nonlinear conservation law is the (inviscid) Burgers' equation

$$u_t + \left(\frac{1}{2} u^2 \right)_x = 0. \quad (1.10)$$

Strong solutions to this problem are given by the implicit equation

$$u(t, x) = u_0(x - u(t, x)t).$$

For non-linear conservation laws, the characteristic speed is a function of the solution itself, and is not constant as in the case of the linear advection (1.9). Distortions may form in the solution as it advances in time, resulting in the crossing of characteristics and thus leading to loss of uniqueness of solutions. At the time t_b where the characteristics first cross, the function $u(t, x)$ has an infinite slope, the wave breaks and a shock forms. In general, entropy satisfying weak solutions may contain shocks or rarefaction waves. Discontinuous solutions for a Cauchy problem can also occur when we have piecewise constant initial data,

$$u(0, x) = \begin{cases} u_l, & x > 0; \\ u_r, & x < 0. \end{cases}$$

The conservation law combined with this type of initial data is called a *Riemann problem* and the form of the solution depends on the relationship between u_l and u_r .

If a shock is formed in the solution of a conservation law (1.8), its speed of propagation is determined by conservation. The relationship between the shock speed s and the states u_l and u_r on either side of the shock is given by the Rankine-Hugoniot jump condition:

$$s = \frac{f(u_l) - f(u_r)}{(u_l - u_r)}. \quad (1.11)$$

1.1.3 Multidimensional conservation laws

Many problems of interest involving conservation laws are solved in more than one space dimension. In d -dimensions ($d > 1$), the conservation law (1.3) can be written in the form

$$u_t + \sum_{i=1}^d f_i(u)_{x_i} = 0, \quad (1.12)$$

with initial data

$$u(0, \mathbf{x}) = u_0(\mathbf{x}), \quad (1.13)$$

where u is a function of $t \in \mathbb{R}^+$ and $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$, and $f_i(u)$, $i = 1, \dots, d$ are the flux functions in the x_i direction. Now, given a solution $u \in C^1(\mathbb{R}^+ \times \mathbb{R}^d)$ of (1.12), we define a characteristic of (1.12) to be a curve $\xi : \mathbb{R} \rightarrow \mathbb{R}^d$ that satisfies the ordinary differential equation

$$\frac{d\xi(t)}{dt} = f'(u(t, \xi(t))) \quad \text{where} \quad f' = (f'_1, \dots, f'_d)^T, \quad (1.14)$$

i.e.

$$\frac{d\xi_i(t)}{dt} = f'_i(u(t, \xi(t))) \quad \text{for each} \quad i = 1, \dots, d. \quad (1.15)$$

For a fixed characteristic ξ , we denote by D_t the differential operator

$$D_t \equiv \partial_t + \sum_{i=1}^d f'_i(u(t, \xi(t))) \partial_{x_i}, \quad (1.16)$$

which is the directional derivative along ξ . Thus, (1.12) can be written as

$$D_t u = 0,$$

i.e. the solution $u \in C^1(\mathbb{R})$ is constant along the characteristics. Now $\frac{d\xi}{dt}$ is constant, i.e. $\xi(t)$ is straight line in \mathbb{R}^d and it follows that

$$u(t, \mathbf{x}) = u_0(\mathbf{x} - t f'(u(t, \mathbf{x}))) \quad \text{where} \quad f' = (f'_1, \dots, f'_d)^T, \quad (1.17)$$

which is an implicit equation for $u(t, \mathbf{x})$. Differentiating with respect to x_j yields

$$\partial_{x_j} u(t, \mathbf{x}) = \frac{\partial_{x_j} u_0(\mathbf{z})}{1 + t \left(\sum_{i=1}^d f''_i(u_0(\mathbf{z})) \partial_{x_i} u_0(\mathbf{z}) \right)}, \quad (1.18)$$

where $\mathbf{z} = \mathbf{x} - t f'(u(t, \mathbf{x}))$. Thus, any solution of (1.12) whose initial data is such that

$$\kappa := \min_{\mathbf{z} \in \mathbb{R}^d} \{ f''_i(u_0(\mathbf{z})) \partial_{x_i} u_0(\mathbf{z}) \} < 0,$$

will suffer gradient blow up in finite time (in at least one of its partial derivatives) at time $t_b = \frac{-1}{\kappa}$.

Existence and uniqueness proofs for admissible solutions of multidimensional conservation laws usually rely upon compactness arguments for sequences of solutions generated by the vanishing viscosity method [74] or low-order finite difference approximations [29]. Moreover, just over twenty years ago, uniqueness results have been generalized using the concept of measure-valued solutions (see DiPerna [32]) providing a new tool for convergence proofs for a variety of numerical methods. However, there are still no

general existence results for multidimensional systems.

1.2 Numerical Methods

A large number of conservation laws are nonlinear and as such their analytical solutions may be impossible to obtain. This has motivated the need to use numerical methods for most practical applications. There are basically three main families of numerical methods used for solving (1.3) - (1.4): finite difference methods, finite volume methods (FVM) and finite element methods (FEM). In order to handle problems on complex geometries and to do local mesh refinement it is preferable to use unstructured grids consisting of triangles, quadrilaterals and other polygons. Moreover, when working on unstructured grids, one would need to use either finite element methods or finite volume methods.

Finite difference methods are the oldest of the methods used in the numerical solution of differential equations. The first application was considered to have been developed by Euler in 1768. Based on Taylor series and on the approximate definition of derivatives, they are simple and straight forward methods for the discretization of differential equations but usually require a high degree of mesh regularity. Examples of finite difference methods for hyperbolic conservation laws include the Lax-Friedrichs method, the Lax-Wendroff method, the leap-frog method and the Beam-Warming method. The order of any finite difference method can usually be obtained via Taylor expansions, and the convergence and stability theory of these methods is well known. Details on the application of finite difference methods to conservation laws can be found in the books of Kröner [74], Hirsch [54] and Morton & Mayers [87]. The main advantage of finite difference methods lies in their ease of implementation. One reason why finite difference methods are not always utilised for the numerical solution of conservation laws is the fact that they always require structured meshes which may not be suitable on certain computational domains or for certain applications.

Traditionally finite element methods are used for the numerical solution of differential equations arising from variational minimization problems where the approximate solution is represented by a finite number of basis functions spanning an appropriate finite dimensional approximation solution space. Over the years, there have been several formulations of the finite element method and the method has been applied to a wide range of problems as well as to all classes of partial differential equations (PDE). Recently, there has been great interest in the design and analysis of discontinuous Galerkin (DG) finite element methods for the discretisation of elliptic, parabolic and hyperbolic PDEs. These methods are based on approximations that are discontinuous across element interfaces, where continuity of boundary element fluxes is weakly enforced. A

detailed survey of the application of the finite element method to conservation laws can be found in [25].

The finite volume method, which is the main subject of this work, is a numerical method for solving partial differential equations that computes the values of the conserved variables averaged across a control volume. In the rest of this section, we will provide a brief survey of the family of finite volume methods.

1.2.1 A survey of finite volume methods

In general, the design of a finite volume method consists of two steps. In the first step, given initial conditions, constant, linear or high order polynomials are defined within the control volume from the cell average values of the variables. The second step involves the interface fluxes of the control volume, from which the cell averages of the variables are then obtained for a solution at the next time level. The flux computation in these methods can be categorised into two types: the *centred* schemes and the *upwind* schemes. Centred schemes are based on the averaging of Riemann fans; a technique usually implemented on staggered grids. Centred schemes require no Riemann solvers. Therefore, all that one has to do in order to solve a problem with such schemes is to supply the flux function.

In upwind methods, a polynomial is reconstructed in each cell and then used in computing a new cell average of the same cell at the next time step. These methods require solving Riemann problems or computing numerical fluxes at the discontinuous interface. The family of Godunov-type methods are generally considered to be the most successful upwind methods for the numerical solution of hyperbolic conservation laws. The original upwind method of Godunov [41] uses piecewise constant data (usually the cell averages) on each cell. This method is only first order accurate and introduces a large amount of numerical diffusion yielding poor accuracy and smeared results. In addition, Godunov [41] has shown that monotonicity preserving linear schemes are at most first order accurate. The low order accuracy of these linear schemes has led to the development of higher order accuracy schemes which make use of nonlinearity, so that both resolution of discontinuities and high order away from discontinuities can be attained. Second order accurate methods such as Fromm's method, Beam-Warming method and Lax-Wendroff method [78] are obtained by using piecewise linear reconstructions on each control volume. These methods give oscillatory approximations to discontinuous solutions as shown in Figure 1.1.

An early high resolution generalisation of the Godunov finite volume method to higher order of accuracy was due to Bram van Leer [134, 135, 136, 137, 138]. In this series of papers, he developed amongst other things an approach known as the MUSCL

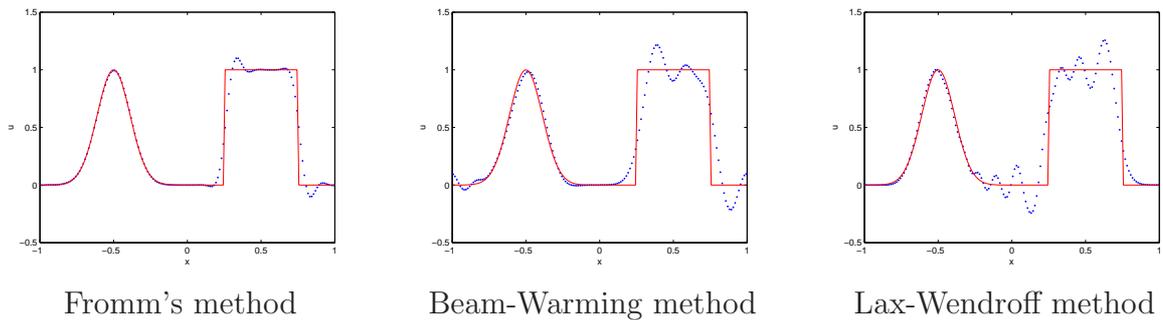


Figure 1.1: Some finite volume methods for the linear advection equation showing oscillations near discontinuities at time $t = 2$ and $N = 160$ gridpoints. The bold line represents the exact solution while the dotted line represents the finite volume approximation.

(Monotone Upstream-centred Scheme for Conservation Laws) method. His method and other high resolution methods are based on linear reconstructions that are able to suppress possible oscillations by using the so-called slope-limiters. Examples of such limiters include the minmod limiter, the superbee limiter, the monotonised central difference limiter and the van Leer limiter. Their ability to suppress oscillations is shown in Figure 1.2. A detailed description of these methods and other finite volume methods can be found

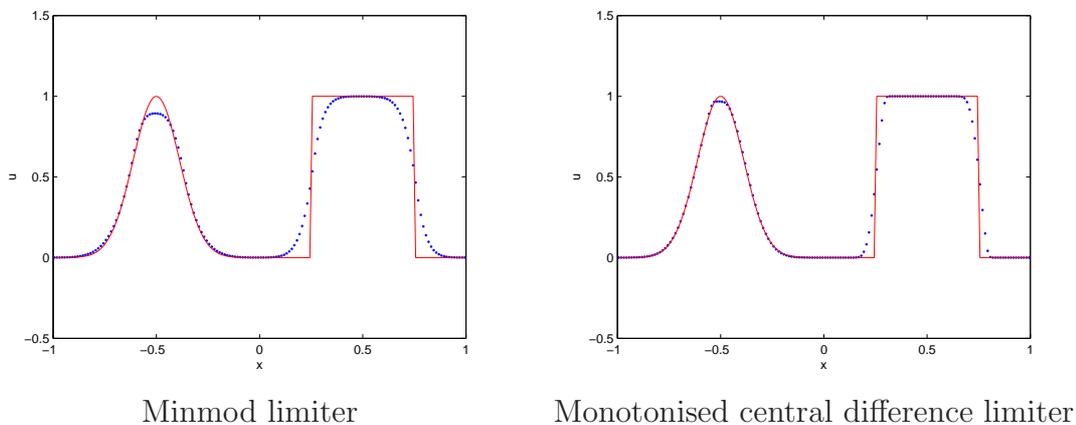


Figure 1.2: High resolution finite volume methods with slope limiters for the linear advection equation at time $t = 2$ and $N = 160$ grid points. The bold line represents the exact solution while the dotted line represents the finite volume approximation.

in the books of Leveque [78], Kroner [74] and Toro [128]. This class of methods satisfy a Total Variation Diminishing (TVD) property and have been analysed by Harten [49] and Osher [92]. A major weakness of slope-limiting methods is that their accuracy inevitably degenerates to first order near discontinuities and even near smooth extrema. In addition, they may produce excessive numerical dissipation. They may therefore be unsuitable for applications involving long time simulations of complex structures like

acoustics and compressible fluid flow. The ideas of van Leer were extended to quadratic approximations by Colella and Woodward [28] in form of the Piecewise Parabolic Method (PPM). Most of these methods, although initially developed for problems in one dimension, have been successfully extended to multidimensional problems, e.g. [27].

Although Godunov's method and its generalisations can also be interpreted in one space dimension as finite difference methods, concepts originally developed in 1D, such as solution monotonicity and discrete maximum principle analysis are often used in the design of finite volume methods in multi-dimensions and on unstructured meshes where finite difference methods are not always suitable [9].

The Essentially Non-Oscillatory (ENO) method was first developed as a finite volume method by Harten *et al* [52] in 1987 and it is perhaps the first successful attempt to obtain a uniformly high order accurate extension of the van Leer approach. In the ENO reconstruction method, the data in each cell can be represented by polynomials of arbitrary order and not just linear or quadratic ones. The numerical solutions obtained by these methods are almost free from spurious oscillations. The reconstruction procedure in [52] is an extension of an earlier reconstruction technique found in [53]. The key idea of the ENO method of order k is to consider an appropriate number of possible stencils covering a given control volume and to select only one, the smoothest, using some appropriate criterion like divided differences in one dimension or some suitable norm in two-dimensions. The reconstruction polynomial is then built using this stencil. Numerical results for the ENO scheme have shown that the method is indeed uniformly high order accurate and resolves shocks with sharp and monotone (to the eye) transitions. It is also worthy to note that a finite difference version of the ENO scheme was developed by Shu & Osher [110, 111]. In the years to follow, there has been a lot of work on improving the methodology and expanding the range of application of the ENO method [2, 15, 50, 109]. The ENO method was later extended to multiple space dimensions on arbitrary meshes by Abgrall [1], Harten and Chakravarthy [51], and Sonar [116].

In recent years, the RK-WENO method has become a popular finite difference and finite volume method for the numerical solution of conservation laws and related equations. It was developed as an improvement of the ENO schemes. The first RK-WENO schemes were constructed for one dimensional conservation laws by Liu, Osher & Chan [81] and Jiang & Shu [68] and were later extended to the two-dimensional setting by Friedrich [38] and Hu & Shu [59]. Furthermore, Titarev and Toro [126] have used a dimension-splitting technique to implement a RK-WENO scheme for three dimensional conservation laws on Cartesian grids. In the WENO framework, the whole set of stencils and their corresponding polynomial reconstructions are used to construct a weighted sum of reconstruction polynomials to approximate the solution over the control volumes

of the finite volume method. Since these early developments, RK-WENO schemes have been used successfully in a wide range of applications and to solve other convection dominated problems. They have been further developed and analysed in [79, 97, 107], have been used to solve balance laws by Vukovic *et al* [139] and have been used in the numerical solution of Hamilton-Jacobi equations [149]. Advantages of RK-WENO schemes over ENO include smoothness of numerical fluxes, better steady state convergence, and generally better accuracy using the same stencils [59].

On Cartesian grids the RK-WENO method can be formulated both as a finite difference method and as a finite volume method. The finite difference formulation of the RK-WENO method is based on a convex combination of fluxes rather than a convex combination of recovery functions. However, on unstructured grids, the method can only be implemented in the finite volume setting.

The ADER-WENO method is a relatively new Godunov-type method for constructing non-oscillatory finite volume schemes for hyperbolic conservation laws which are of arbitrary high-order in space and time for smooth problems and with optimal stability conditions for all problems. It is a fully discrete finite volume method that combines high order WENO reconstruction from cell averages with high order flux evaluation. It was first developed in 2001 for linear advection problems with constant coefficients by Toro *et al* [129, 130]. The ADER-WENO method has been utilised in [73, 105, 125, 127, 131, 132, 133] for the solution of both scalar conservation laws and systems of conservation laws in one and several space dimensions on structured and unstructured meshes. Note that all the known ADER-WENO methods in the literature are based on polynomial reconstruction methods.

1.3 Objectives, Outline and Main Results

1.3.1 Objectives and motivation

This thesis focuses on the design and implementation of non-oscillatory finite volume methods for conservation laws on unstructured triangular grids. There are several challenges one would face when developing such methods. These include conservation in the presence of shock waves, and the fact that spurious oscillations may be generated in the vicinity of shock waves. Any useful numerical method for solving conservation law must seek to resolve these difficulties. In addition, the hyperbolic nature of the governing equations and the presence of solution discontinuities makes high order difficult to attain. As a result, several large scale applications still use low order methods, even though there is substantial numerical evidence indicating that the high order methods may offer a way to significantly improve the resolution and quality of these computa-

tions [24, 28, 59, 133].

There are currently several non-oscillatory numerical discretisations that have been constructed for achieving high resolution and high order accuracy away from discontinuities but we will focus on two in this thesis: the RK-WENO method and the ADER-WENO method. These two methods are well known in the literature for the numerical solution of hyperbolic conservation laws and are usually based on specially designed polynomial reconstruction methods.

In this work, our goal is to use *polyharmonic splines*, a class of Radial Basis Functions (RBFs), as an alternative basis for reconstruction in order to achieve high order in space in the WENO reconstruction algorithm. We will particularly focus on the use of the popular *thin plate splines* which are the two-dimensional analogue of the one dimensional cubic spline. The use of radial basis functions is motivated by the fact that it is suitable for reconstruction on both structured and unstructured meshes. Radial basis functions can also be effectively implemented on complex computational domains and are suitable for interpolation and reconstruction in arbitrary dimensions. Moreover due to its radial nature, if a basis function is suitable for reconstruction in d dimensions, it can generally be used in any dimension less than d [36]. This allows theory and algorithms to be implemented in low space dimensions with straight forward extensions to problems in higher space dimensions. To this end, even though the methods we develop and implement in this work are in two space dimensions, we believe they can be used for the numerical solution of conservation laws in higher space dimensions. Another motivation for using RBFs lies in the fact that the linear systems associated with RBF interpolation are guaranteed to be invertible under very mild conditions on the location of the data points or geometry of the grid. RBF interpolants also possess a number of optimality properties in their associated native spaces. More specifically, polyharmonic spline reconstruction technique is numerically stable, flexible and of arbitrary high local approximation order [63]. Moreover, due to the theory of polyharmonic splines, optimal reconstructions are obtained in the associated native spaces, the Beppo-Levi spaces.

We will also seek to design adaptive algorithms using the RK-WENO and ADER-WENO methods to harness the benefits of adaptivity: the reduction of computational cost coupled with improved accuracy. We will implement three types of adaptive strategies: stencil adaptivity, mesh adaptivity, and mesh & order adaptivity. We also wish to show that our methods provide competitive results when used in solving both linear and nonlinear conservation laws that arise in a number of applications.

1.3.2 Main results

The main results of this thesis are detailed below.

1. Approximation order and numerical stability of local reconstruction by polyharmonic splines.

- Convergence rates for local interpolation from cell averages using polyharmonic splines are presented. A relationship between the derivatives of the Lagrange basis functions for polyharmonic spline interpolation is established. This extends the earlier results in [63]. An algorithm for the stable evaluation of the derivatives of the polyharmonic spline interpolant is provided.

2. The RK-WENO method based on polyharmonic spline reconstruction.

- The RK-WENO method where the local reconstruction step is performed using polyharmonic splines is proposed and implemented in combination with SSP Runge-Kutta time stepping. Numerical results and convergence rates are provided which agree with expected theoretical results.
- Numerical investigations and recommendations on suitable stencil sizes for polyharmonic spline reconstruction are provided.
- The suitability of the Beppo-Levi norm as an oscillation indicator for polyharmonic spline reconstruction is also established.

3. The ADER-WENO method using the polyharmonic spline reconstruction.

- The WENO reconstruction based on polyharmonic splines is used in the reconstruction step of the ADER-WENO method.
- The high order flux evaluation of the ADER-WENO method is achieved using the polyharmonic spline interpolant and its derivatives as the piecewise initial data of a set of Generalised Riemann Problems.
- The ADER-WENO method with this new RBF reconstruction technique is implemented and supporting numerical results are provided.

4. Adaptivity.

- A simple stencil adaptivity strategy is implemented to reduce computational effort. This is based on the flexibility in the choice of stencil sizes in radial basis function reconstruction. This stencil adaptivity strategy is also implemented in combination with mesh adaptivity.
- Mesh adaptivity is successfully coupled with the RK-WENO and ADER-WENO methods using a suitable *a posteriori* error indicator.

- Some results on mesh & order adaptivity are presented. This reveals a reduction in the number of degrees of freedom used in the computations while delivering results of good accuracy. To the best of our knowledge, these are the first multidimensional mesh and order adaptive computations for finite volume methods available in the literature.

1.3.3 Outline of the thesis

The rest of the thesis is structured as follows.

- In Chapter 2 we will begin by giving a survey of the definition and properties of radial basis functions which are powerful tools for scattered data approximation. This class of functions are the main tool we will use in designing the numerical methods in this thesis. Next, we will introduce the concept of generalised interpolation and focus on the situation where our functionals are cell average operators which is the relevant kind of interpolation for our purposes. We will provide an error estimate for reconstruction from cell averages with thin plate splines in particular and for polyharmonic splines in general.

Since the finite volume methods we are going to implement are based on local reconstruction methods, we will present some existing results on the approximation order and numerical stability of local generalized interpolation by polyharmonic splines. We will provide new results on the stable evaluation of the derivatives of the polyharmonic spline interpolant.

- In Chapter 3 we will give a detailed algorithmic description of the RK-WENO method. We describe the polyharmonic spline WENO reconstruction method and discuss other key ingredients of the method like time stepping and stencil selection. We will show numerical results for standard test cases. We will also apply the RK-WENO method to Doswell's frontogenesis, a challenging problem with a velocity field that is a steady circular vortex which leads to a solution with multiscale behaviour.
- In Chapter 4 we describe the ADER-WENO method and present the formulation of the method using the polyharmonic spline WENO reconstruction of Chapter 3. We will also present several numerical results to validate our proposed ADER-WENO method using standard test problems. The robustness of the method will be verified using Smolarkiewicz's deformational test.
- In Chapter 5 we consider adaptive algorithms using the methods developed in Chapters 3 and 4 as a technique for improving accuracy and reducing computational cost. We present several numerical examples of problems solved with the

adaptive versions of the RK-WENO method and ADER-WENO method. Results of the application of the adaptive methods to a problem with time dependent velocity fields and to the simulation of two-phase flow in porous media will be included.

- In Chapter 6, mesh & order adaptivity which combines mesh refinement with order variation procedures is investigated and preliminary results are presented.
- The final chapter draws some conclusions and gives an outlook of further research directions.

Chapter 2

Radial Basis Functions

2.1 Radial Basis Function Interpolation

In certain applications, a function u may not be given as a formula but as a set of function values. These data may take the form of exact or approximate values of u at some scattered points in the domain $\Omega \in \mathbb{R}^d$ of definition of u . In general, a recovery problem involves the reconstruction of u as a formula from the given set of function values. The recovery of u may be done either by interpolation, which tries to match the data exactly, or by approximation, which allows u to miss some or all of the data in some way. The decision on whether to use interpolation or approximation usually depends on the application, the choice of the function spaces and what properties the recovery process is required to satisfy [101].

Radial Basis Functions (RBFs) are well-established and efficient tools for the multi-variate interpolation of scattered data. They are the primary tool used in this work in the reconstruction step of the spatial discretisation of the finite volume method. In the past two decades, radial basis functions have been used extensively in the numerical solution of partial differential equations. In particular, RBFs have been used in collocation methods for elliptic equations [37], transport equations [82] and the equations of fluid dynamics [69]. RBFs have also been used in the theory of meshfree Galerkin methods by Wendland [142], in semi-Lagrangian methods for advection problems by Behrens & Iske [12], in meshfree methods for advection-dominated diffusion problems in the thesis of Hunt [61], and also in the recovery step of finite volume ENO schemes [67, 115].

In this work, we employ local interpolation with RBFs in the WENO reconstruction step of finite volume discretisations. This yields numerical methods that are of high order, stable, flexible, easy to implement and suitable on both structured and unstructured grids. We also provide a clear analysis of the approximation order and numerical stability of the reconstruction method.

In this section, we present a brief survey of the commonly used radial basis functions

and some of their important properties. Further details on RBF interpolation can be found in [19, 64, 95, 144].

2.1.1 The interpolation problem

Given a vector $u|_X = (u(\mathbf{x}_1), \dots, u(\mathbf{x}_n))^T \in \mathbb{R}^n$ of function values, obtained from an unknown function $u : \mathbb{R}^d \rightarrow \mathbb{R}$ at a finite scattered point set $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{R}^d$, $d \geq 1$, scattered data interpolation requires computing an appropriate interpolant $s : \mathbb{R}^d \rightarrow \mathbb{R}$ satisfying $s|_X = u|_X$, i.e.

$$s(\mathbf{x}_j) = u(\mathbf{x}_j) \quad \text{for all } 1 \leq j \leq n. \quad (2.1)$$

The radial basis function interpolation method utilises a fixed radial function $\phi : [0, \infty) \rightarrow \mathbb{R}$, so that the interpolant s in (2.1) has the form

$$s(\mathbf{x}) = \sum_{j=1}^n c_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) + p(\mathbf{x}), \quad p \in \mathcal{P}_m^d, \quad (2.2)$$

where $\|\cdot\|$ is the Euclidean norm on \mathbb{R}^d and \mathcal{P}_m^d denotes the vector space of all real-valued polynomials in d variables of degree at most $m - 1$, where $m \equiv m(\phi)$ is known as the order of the basis function ϕ . Possible choices for ϕ are, along with their order m , shown in Table 2.1.

RBF	$\phi(r)$	Parameters	Order
Polyharmonic Splines	r^{2k-d} for d odd $r^{2k-d} \log(r)$ for d even	$k \in \mathbb{N}, k > d/2$ $k \in \mathbb{N}, k > d/2$	k k
Gaussians	$\exp(-r^2)$		0
Multiquadrics	$(1 + r^2)^\nu$	$\nu > 0, \nu \notin \mathbb{N}$	$\lceil \nu \rceil$
Inverse Multiquadrics	$(1 + r^2)^\nu$	$\nu < 0$	0

Table 2.1: Radial basis functions (RBFs) and their orders.

Radial basis function interpolants have the nice property of being invariant under all Euclidean transformations (i.e. translations, rotations and reflections). This is because Euclidean transformations are characterized by orthogonal transformation matrices and are therefore Euclidean-norm-invariant [36].

Radial basis functions, like Gaussians, (inverse) multiquadrics, and polyharmonic splines are all globally supported on \mathbb{R}^d . More recently, a class of compactly supported radial basis functions of order 0 ($p(\mathbf{x}) \equiv 0$ in (2.2)) have been constructed, see [18, 141,

146]. While the RBFs in Table 2.1 can be used in any space dimension, the suitability of the compactly supported RBFs depends on the space dimension d .

2.1.2 Solving the interpolation problem

All the basis functions in Table 2.1 (and several others not mentioned here) can be classified using the concept of (conditionally) positive definite functions which can be used in analyzing the existence and uniqueness of the solution of interpolation problems.

Definition 2.1 A continuous radial function $\phi : [0, \infty) \rightarrow \mathbb{R}$ is said to be positive definite on \mathbb{R}^d , if and only if for any finite set $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, $X \subset \mathbb{R}^d$, the $n \times n$ matrix

$$A = ((\phi(\|\mathbf{x}_i - \mathbf{x}_j\|))_{1 \leq i, j \leq n} \in \mathbb{R}^{n \times n}$$

is positive definite.

Definition 2.2 A continuous radial function $\phi : [0, \infty) \rightarrow \mathbb{R}$ is said to be conditionally positive definite of order m on \mathbb{R}^d , if and only if for any finite set $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, $X \subset \mathbb{R}^d$, and all $\mathbf{c} \in \mathbb{R}^n \setminus \{0\}$ satisfying

$$\sum_{j=1}^n c_j p(\mathbf{x}_j) = 0 \tag{2.3}$$

for all $p \in \mathcal{P}_m^d$ the quadratic form

$$\sum_{j=1}^n \sum_{k=1}^n c_j c_k \phi(\|\mathbf{x}_j - \mathbf{x}_k\|) \tag{2.4}$$

is positive. The function ϕ is positive definite if it is conditionally positive definite of order $m = 0$.

When $m = 0$, the interpolant s in (2.2) has the form

$$s(\mathbf{x}) = \sum_{j=1}^n c_j \phi(\|\mathbf{x} - \mathbf{x}_j\|). \tag{2.5}$$

Using the interpolation conditions (2.1), the coefficients $\mathbf{c} = (c_1, \dots, c_n)^T \in \mathbb{R}^n$ of s in (2.5) can be obtained by solving the linear system

$$A\mathbf{c} = u|_X, \tag{2.6}$$

where $A = ((\phi(\|\mathbf{x}_i - \mathbf{x}_j\|))_{1 \leq i, j \leq n} \in \mathbb{R}^{n \times n}$. From Definition 2.1, the matrix A is positive definite provided ϕ is positive definite. An important property of positive definite ma-

trices is that all their eigenvalues are positive, and therefore a positive definite matrix is non-singular. Hence, the system (2.6) has a unique solution provided ϕ is positive definite. Moreover, for $m = 0$, the interpolation problem has a unique solution s of the form (2.5) if ϕ is positive definite [64].

For $m > 0$, ϕ is conditionally positive definite of order m and the interpolant s in (2.2) contains a nonzero polynomial part, yielding q additional degree of freedom, where $q = \binom{m-1+d}{d}$ is the dimension of the polynomial space \mathcal{P}_m^d . These additional degrees of freedom are usually eliminated using the q vanishing moment conditions

$$\sum_{j=1}^n c_j p(\mathbf{x}_j) = 0, \quad \text{for all } p \in \mathcal{P}_m^d. \quad (2.7)$$

In total, this amounts to solving the $(n + q) \times (n + q)$ linear system

$$\begin{bmatrix} A & P \\ P^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} u|_X \\ 0 \end{bmatrix}, \quad (2.8)$$

where $A = ((\phi(\|\mathbf{x}_i - \mathbf{x}_j\|)))_{1 \leq i, j \leq n} \in \mathbb{R}^{n \times n}$, $P = ((\mathbf{x}_j)^\alpha)_{1 \leq j \leq n; |\alpha| < m} \in \mathbb{R}^{n \times q}$, and $\mathbf{d} = (d_\alpha)_{|\alpha| < m} \in \mathbb{R}^q$ for the coefficients of the polynomial part in (2.2).

The linear system (2.8) has a unique solution for the unknown coefficients $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{d} \in \mathbb{R}^q$, provided the set of interpolation points $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ is \mathcal{P}_m^d -unisolvent, i.e., for $p \in \mathcal{P}_m^d$ we have

$$p(\mathbf{x}_j) = 0 \quad \text{for } 1 \leq j \leq n \quad \Rightarrow \quad p \equiv 0.$$

In this case, any polynomial in \mathcal{P}_m^d can uniquely be reconstructed from its function values sampled at the points in X .

The following theorem summarizes the connection between conditional positive definiteness and the existence of a unique solution for the interpolation problem.

Theorem 2.3 ([144]) *Suppose ϕ is conditionally positive definite of order m on \mathbb{R}^d . Suppose further that the set $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, $X \subset \mathbb{R}^d$ is \mathcal{P}_m^d unisolvent. Then there is exactly one function s of the form (2.2) such that*

$$s(\mathbf{x}_j) = u(\mathbf{x}_j), \quad 1 \leq j \leq n \quad \text{and} \quad \sum_{j=1}^n c_j p(\mathbf{x}_j) = 0, \quad \text{for all } p \in \mathcal{P}_m^d.$$

2.1.3 Characterization of conditionally positive definite functions

It is clear that interpolation with radial basis functions relies on the conditional positive definiteness of the chosen basis function ϕ . To this end, we briefly review the characterisation of conditionally positive definite functions using the concept of *completely monotone functions*.

The question of whether or not ϕ is conditionally positive definite of some order m on \mathbb{R}^d was answered by Schoenberg [103] for positive definite functions (i.e. $m = 0$) in terms of completely monotone functions. The sufficient part of Schoenberg's result was extended to conditionally positive definite functions by Micchelli [85] who also conjectured the necessity of this criterion. This conjecture was proved some years later by Guo, Hu and Sun [46].

Definition 2.4 (Completely monotone function) *A function f is said to be completely monotone on $(0, \infty)$ if $f \in C^\infty(0, \infty)$ and $(-1)^k f^{(k)}$ is non-negative for all $k \in \mathbb{N}_0$.*

Theorem 2.5 (Micchelli [85]) *Given a function $\phi \in C^\infty(0, \infty)$, define $f = \phi(\sqrt{\cdot})$. If there exists an $m \in \mathbb{N}_0$ such that $(-1)^m f^{(m)}$ is well-defined and completely monotone but not identically constant on $(0, \infty)$, then ϕ is conditionally positive definite of order m on \mathbb{R}^d for all $d \geq 1$.*

Theorem 2.5 allows us to show that any ϕ in Table 2.1 is conditionally positive definite of order m . We illustrate this using two examples from [144].

Example 2.1 *The functions $\phi(r) = (-1)^{\lceil k/2 \rceil} r^k$, where k is an odd number, are conditionally positive definite of order $m \geq \lceil k/2 \rceil$ on \mathbb{R}^d for all $d \geq 1$.*

Define $f_k(r) = (-1)^{\lceil k/2 \rceil} r^{\frac{k}{2}}$ to get

$$f_k^{(\ell)}(r) = (-1)^{\lceil k/2 \rceil} \frac{k}{2} \left(\frac{k}{2} - 1 \right) \cdots \left(\frac{k}{2} - \ell + 1 \right) r^{\frac{k}{2} - \ell}.$$

This shows that $(-1)^{\lceil k/2 \rceil} f_k^{\lceil k/2 \rceil}(r)$ is completely monotone and $m = \lceil \frac{k}{2} \rceil$ is the smallest possible choice.

Example 2.2 *The functions $\phi(r) = (-1)^{k+1} r^{2k} \log(r)$ are conditionally positive definite of order $m = k + 1$ on \mathbb{R}^d .*

Since $2\phi(r) = (-1)^{k+1} r^{2k} \log(r^2)$ we set $f_k(r) = (-1)^{k+1} r^k \log(r)$. Then

$$f_k^{(\ell)}(r) = (-1)^{k+1} k(k-1) \cdots (k-\ell+1) r^{k-\ell} \log(r) + p_\ell(r), \quad 1 \leq \ell \leq k,$$

where p_ℓ is a polynomial of degree $k - \ell$. This then means

$$f_k^{(k)}(r) = (-1)^{k+1} k! \log(r) + c,$$

and finally $f_k^{(k+1)}(r) = (-1)^{k+1} k! r^{-1}$ which is completely monotone on $(0, \infty)$ and so ϕ is conditionally positive-definite of order $k + 1$.

2.1.4 Lagrange form of the interpolant

Sometimes it is more convenient to work with the *Lagrange form*

$$s(\mathbf{x}) = \sum_{j=1}^n \ell_j(\mathbf{x}) u(\mathbf{x}_j) \quad (2.9)$$

of the interpolant s in (2.2), where the *Lagrange basis functions* (also known as the *cardinal basis functions*) $\ell_1(\mathbf{x}), \dots, \ell_n(\mathbf{x})$ satisfy

$$\ell_j(\mathbf{x}_k) = \begin{cases} 1, & \text{for } j = k, \\ 0, & \text{for } j \neq k, \end{cases} \quad 1 \leq j, k \leq n, \quad (2.10)$$

and so $s(\mathbf{x}_j) = u(\mathbf{x}_j)$, $j = 1, \dots, n$. For radial basis function approximation, this idea is due to Wu & Schaback [147]. Moreover, this representation exists for all conditionally positive definite functions, see [36, 144]. For a point $\mathbf{x} \in \mathbb{R}^d$, the vectors $\ell(\mathbf{x}) = (\ell_1(\mathbf{x}), \dots, \ell_n(\mathbf{x}))^T$ and $v(\mathbf{x}) = (v_1(\mathbf{x}), \dots, v_q(\mathbf{x}))^T$, $q = \dim(\mathcal{P}_m^d)$, are the unique solution of the linear system

$$\mathbf{A}v(\mathbf{x}) = \beta(\mathbf{x}) \quad (2.11)$$

where

$$\mathbf{A} = \begin{bmatrix} A & P \\ P^T & \mathbf{0} \end{bmatrix}, \quad v(\mathbf{x}) = \begin{bmatrix} \ell(\mathbf{x}) \\ v(\mathbf{x}) \end{bmatrix}, \quad \beta(\mathbf{x}) = \begin{bmatrix} R(\mathbf{x}) \\ S(\mathbf{x}) \end{bmatrix}$$

and

$A = ((\phi(\|\mathbf{x}_i - \mathbf{x}_j\|))_{1 \leq i, j \leq n} \in \mathbb{R}^{n \times n}$, $R(\mathbf{x}) = \phi(\|\mathbf{x} - \mathbf{x}_j\|)_{1 \leq j \leq n}$, $S(\mathbf{x}) = (\mathbf{x}^\alpha)_{|\alpha| < m} \in \mathbb{R}^q$.

Combining the representations of s in (2.2) and (2.9) yields

$$\begin{aligned} s(\mathbf{x}) &= \langle \ell(\mathbf{x}), u|_X \rangle = \langle v(\mathbf{x}), u_X \rangle \\ &= \langle \mathbf{A}^{-1} \beta(\mathbf{x}), u_X \rangle = \langle \beta(\mathbf{x}), \mathbf{A}^{-1} u_X \rangle \\ &= \langle \beta(\mathbf{x}), \mathbf{b} \rangle, \end{aligned} \quad (2.12)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product of the Euclidean space \mathbb{R}^d , and where we have set

$$u_X = \begin{bmatrix} u|_X \\ 0 \end{bmatrix} \in \mathbb{R}^{n+q} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} \in \mathbb{R}^{n+q}$$

for the right hand side and the solution of the linear system (2.8).

2.1.5 The optimality of RBF interpolation

Each conditionally positive definite function ϕ is associated with a native Hilbert space \mathcal{N}_ϕ equipped with a semi-norm $|\cdot|_\phi$ in which it solves an *optimal recovery* problem. This means that for any $u \in \mathcal{N}_\phi$ and $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, the unique RBF interpolant s of the form (2.2) lies in the native space \mathcal{N}_ϕ and satisfies:

$$|s|_\phi = \min\{|\tilde{s}|_\phi : \tilde{s} \in \mathcal{N}_\phi \quad \text{with} \quad \tilde{s}(\mathbf{x}_j) = u(\mathbf{x}_j), \quad 1 \leq j \leq n\}. \quad (2.13)$$

The theory of optimal recovery of interpolants was first described in the late 1950s by Golomb & Weinberger [42] and later studied in detail by Micchelli & Rivlin [86].

2.1.6 Numerical stability

To proceed with our discussion, we first of all need to define two important quantities.

Definition 2.6 *Given a finite set $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \Omega$ of pairwise distinct points, the fill distance of X is given as*

$$h_{X,\Omega} = \sup_{\mathbf{y} \in \Omega} \min_{\mathbf{x} \in X} \|\mathbf{x} - \mathbf{y}\|, \quad (2.14)$$

while the separation distance or packing radius of X is defined as

$$q_X = \min_{\mathbf{x}, \mathbf{y} \in X, \mathbf{x} \neq \mathbf{y}} \|\mathbf{x} - \mathbf{y}\|. \quad (2.15)$$

Numerical stability is usually a very important aspect of any interpolation scheme. We particularly need to be sure that, as we refine a set of interpolation points (i.e. as the fill distance $h_{X,\Omega}$ tends to zero), the method does not become numerically unstable. A standard criterion for measuring the numerical stability of an interpolation process is the condition number of the interpolation matrix. In particular, for radial basis function interpolation, we need to examine the condition number of the matrix on the left hand side of the linear system (2.8). The condition number of any matrix is given by $\kappa_{\max}/\kappa_{\min}$ where κ_{\max} and κ_{\min} are the maximum and minimum eigenvalues of the matrix and so numerical stability requires that we keep this ratio small. For the RBF interpolation

matrix, there are several upper bounds for κ_{\max} in the literature, and numerical tests show that it indeed causes no problem [100, 102, 144]. However, κ_{\min} is a function of the separation distance of the set X and tends to zero and so may generally spoil the stability of the interpolation process. Thus, the results on numerical stability in the literature focus on lower bounds for κ_{\min} .

Indeed, for every basis function ϕ , there is a function G_ϕ such that

$$\kappa_{\min} \geq G_\phi(q_X).$$

$G_\phi : [0, \infty) \rightarrow [0, \infty)$ is a continuous and monotonically increasing function with $G_\phi(0) = 0$. The form of G_ϕ for various RBFs can be found in [100, 102, 144]. For example, when $\phi(r) = r^k$, $G_\phi(q) = q^k$ and when $\phi(r) = r^{2k} \log(r)$, $G_\phi(q) = q^{2k}$. In both cases, the lower bound goes to zero with decreasing separation distance.

In general, the matrices arising from RBF interpolation may become very ill-conditioned as the minimal separation distance q_X of X is reduced. Thus, to prevent numerical instability in the practical implementation of an RBF interpolation scheme, a preconditioner may be required particularly as the separation distance gets smaller. To this end, in Chapter 3, we will implement a preconditioner for local interpolation with polyharmonic splines, which will indeed be proven to be very relevant in practice.

2.1.7 Polyharmonic splines

Polyharmonic splines, also referred to as *surface splines*, are a special family of radial basis functions. They are particularly useful because of the explicit knowledge of the native space where they solve the optimal recovery problem and the fact that their conditioning is invariant under scalings. The theory of polyharmonic splines as a powerful tool for multivariate interpolation was developed by Duchon [34] in the 1970s. A few years later, Meinguet [83] established a clear framework for using polyharmonic splines as a practical tool for multivariate interpolation. The polyharmonic spline interpolation scheme uses the fixed radial function

$$\phi_{d,k}(r) = \begin{cases} r^{2k-d} \log(r), & \text{for } d \text{ even;} \\ r^{2k-d}, & \text{for } d \text{ odd,} \end{cases} \quad (2.16)$$

where k is required to satisfy $2k > d$ and the order is $m = k$. The interpolant then has the form

$$s(\mathbf{x}) = \sum_{j=1}^n c_j \phi_{d,k}(\|\mathbf{x} - \mathbf{x}_j\|) + p(\mathbf{x}), \quad p \in \mathcal{P}_k^d. \quad (2.17)$$

The polyharmonic splines can be seen as a generalisation of the univariate cubic splines to a multidimensional setting and $\phi_{d,k}$ is the fundamental solution to the iterated Laplacian,

i.e.

$$\Delta^k \phi_{d,k}(\|\mathbf{x}\|) = c\delta_{\mathbf{x}},$$

in the sense of distributions. For $k = d = 2$, we have the thin plate spline $\phi_{2,2}(r) = r^2 \log(r)$ which is the fundamental solution of the biharmonic equation, i.e.,

$$\Delta^2 \phi_{2,2}(\|\mathbf{x}\|) = c\delta_{\mathbf{x}}.$$

The native space of the polyharmonic splines are the *Beppo-Levi spaces* which are defined as follows.

Definition 2.7 For $k > d/2$, the linear space

$$\text{BL}_k(\mathbb{R}^d) := \{u \in C(\mathbb{R}^d) : D^\alpha u \in L_2(\mathbb{R}^d) \text{ for all } |\alpha| = k\}$$

equipped with the inner product

$$(u, v)_{\text{BL}_k(\mathbb{R}^d)} := \sum_{|\alpha|=k} \frac{k!}{\alpha!} (D^\alpha u, D^\alpha v)_{L_2(\mathbb{R}^d)}$$

is called the *Beppo-Levi space* on \mathbb{R}^d of order k .

This means that for a fixed finite point set $X \subset \mathbb{R}^d$, an interpolant s in (2.17) minimises

$$|u|_{\text{BL}_k(\mathbb{R}^d)}^2 = \int_{\mathbb{R}^d} \sum_{|\alpha|=k} \binom{k}{\alpha} (D^\alpha u)^2 d\mathbf{x}, \quad (2.18)$$

among all the functions u of the Beppo-Levi space satisfying $u|_X = s|_X$. For thin plate splines we have

$$|u|_{\text{BL}_2(\mathbb{R}^2)}^2 = \int_{\mathbb{R}^2} \left(\frac{\partial^2 u}{\partial x_1^2} \right)^2 + 2 \left(\frac{\partial^2 u}{\partial x_1 \partial x_2} \right)^2 + \left(\frac{\partial^2 u}{\partial x_2^2} \right)^2 dx_1 dx_2, \quad \text{for } u \in \text{BL}_2(\mathbb{R}^2), \quad (2.19)$$

where we let x_1 and x_2 denote the two coordinates of $\mathbf{x} = (x_1, x_2)^T \in \mathbb{R}^2$.

The Beppo-Levi spaces are related to the *Sobolev spaces* [3]. In fact, the intersection of all Beppo-Levi spaces $\text{BL}_k(\mathbb{R}^d)$ of order $k \leq m$ yields the Sobolev space $W_2^m(\mathbb{R}^d)$. The Beppo-Levi spaces are sometimes referred to as *homogeneous* Sobolev spaces of order k .

2.2 Generalized Interpolation

In certain applications, like the numerical solution of partial differential equations and financial engineering, it is sometimes necessary to recover a function from other types of

data associated with the function rather than point evaluations. For example, the value of the derivatives of the function at certain points may be known, but not the values of the function itself. Fortunately, the RBF ansatz can be extended to several other more general observation functionals. This also fits into the setting of minimum norm generalized interpolation.

We present this in the framework of Hilbert spaces as follows. Let \mathcal{H} be a Hilbert space and denote its dual by \mathcal{H}' . If $\Xi = \{\lambda_1, \dots, \lambda_n\} \subseteq \mathcal{H}'$ is a set of linearly independent functionals on \mathcal{H} and $u_1, \dots, u_n \in \mathbb{R}$ are certain given values associated with u , then a generalized interpolation problem seeks to find a function $s \in \mathcal{H}$ such that

$$\lambda_i(s) = \lambda_i(u), \quad i = 1, \dots, n \quad \text{where} \quad \lambda_i(u) = u_i, \quad i = 1, \dots, n.$$

The interpolant s is referred to as the *generalized interpolant* and the optimal recovery problem in this setting searches for an interpolant $s \in \mathcal{H}$ such that

$$\|s\|_{\mathcal{H}} = \min\{\|\tilde{s}\|_{\mathcal{H}} : \tilde{s} \in \mathcal{H}, \lambda_i(\tilde{s}) = u_i, i = 1, \dots, n\}.$$

In particular, the generalized RBF interpolant has the form

$$s(\mathbf{x}) = \sum_{j=1}^n c_j \lambda_j^y \phi(\|\mathbf{x} - \mathbf{y}\|) + p(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d \quad \text{and} \quad p \in \mathcal{P}_m^d$$

where the notation λ_j^y indicates the action of the functional λ_j on ϕ viewed as a function of the argument \mathbf{y} . We require the interpolant to satisfy

$$\lambda_i^x(s) = \lambda_i^x(u), \quad i = 1, \dots, n, \quad (2.20)$$

where λ_i^x indicates the action of the functional λ_i on s and u which are treated as functions of \mathbf{x} . To eliminate any additional degrees of freedom, the additional constraints

$$\sum_{j=1}^n c_j \lambda_j^x(p) = 0 \quad \text{for all} \quad p \in \mathcal{P}_m^d,$$

need to be satisfied. This results in the linear system

$$\begin{bmatrix} A & P \\ P^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} u|_{\Xi} \\ 0 \end{bmatrix}, \quad (2.21)$$

where $A = (\lambda_i^x \lambda_j^y \phi(\|\mathbf{x} - \mathbf{y}\|))_{1 \leq i, j \leq n} \in \mathbb{R}^{n \times n}$, $P = (\lambda_j^x(\mathbf{x}^\alpha))_{1 \leq j \leq n, 0 \leq |\alpha| < m} \in \mathbb{R}^{n \times q}$, $q = \dim(\mathcal{P}_m^d)$, and $u|_{\Xi} = (\lambda_i^x(u))_{1 \leq i \leq n} \in \mathbb{R}^n$.

The functionals also allow a Lagrange-type representation

$$s(\mathbf{x}) = \sum_{j=1}^n \ell_j(\mathbf{x}) \lambda_j^x(u),$$

with certain cardinal functions $\ell_j(\mathbf{x})$.

Furthermore, to guarantee that the interpolation problem (2.20) has a unique solution s , we need to generalize the notion of the \mathcal{P}_m^d -unisolvency of an interpolation set. To this end, we insist that the set of linear functionals in Ξ satisfy the following condition

$$\lambda_i^x(p) = 0 \quad \text{for } i = 1, \dots, n \quad \Rightarrow \quad p \equiv 0$$

i.e. any polynomial from \mathcal{P}_m^d can be uniquely reconstructed from its values $\{\lambda_i^x(p)\}_{i=1}^n$. Thus, we can say that the set of functionals Ξ is \mathcal{P}_m^d -unisolvent.

In generalized interpolation, the functionals in Ξ may, for instance, be differential operators as in the case of Hermite-Birkhoff interpolation [62], local integrals as in the case of integral interpolation [11], or cell average operators [115].

2.2.1 Error estimates for reconstruction from cell averages

We now turn to the case where the linearly independent functionals in Ξ are cell average operators. This situation arises in the recovery step of finite volume methods where point values of the unknown solution of a PDE have to be reconstructed from cell average data, e.g. [67, 115]. This is one of the main themes of this work.

If we divide a region $\Omega \in \mathbb{R}^2$ into non-overlapping subregions $\mathcal{T} = \{V_j\}$, then for some integrable function u , the *cell average operators* are defined as

$$\lambda_j^x(u) := \bar{u}_j = \frac{1}{|V_j|} \int_{V_j} u(\mathbf{x}) \, d\mathbf{x}.$$

We first focus on a pointwise error estimate of thin plate spline reconstruction on triangular meshes. Based on the earlier work of Powell [96] and Gutzmer [47], we present a pointwise error estimate for thin plate spline interpolation for situations where interpolation data are cell averages on a triangular mesh. In [96], the results were provided for interpolation of scattered point values while Gutzmer [47] treated the instance where the interpolation data were cell averages on Cartesian grids.

Let $u : \mathbb{R}^2 \rightarrow \mathbb{R}$ be an integrable function. Then the thin plate spline interpolant s subject to the conditions $\lambda_i^x(s) = \lambda_i^x(u)$, $i = 1, \dots, n$, has the form

$$s(\mathbf{x}) = \sum_{i=1}^n c_i \lambda_i^y(\|\mathbf{x} - \mathbf{y}\|^2 \log(\|\mathbf{x} - \mathbf{y}\|)) + d_1 + d_2 x_1 + d_3 x_2, \quad (2.22)$$

where $\mathbf{x} = (x_1, x_2)^T$ and $\mathbf{y} = (y_1, y_2)^T$.

We first of all state without proof the following lemma.

Lemma 2.8 ([96, 47]) *Let λ_i^x , $i = 0, \dots, n$ be a set of $n > 3$ functionals with compact support and unisolvent on \mathcal{P}_2^2 . If*

$$\sum_{i=0}^n \hat{\beta}_i = 0 \quad \text{and} \quad \sum_{i=0}^n \hat{\beta}_i \lambda_i^x(p) = 0 \quad \text{for all } p \in \mathcal{P}_2^2, \quad (2.23)$$

then the functional $\hat{L} = \sum_{i=0}^n \hat{\beta}_i \lambda_i^x$ can be bounded as follows

$$|\hat{L}g| \leq \left[8\pi \|g\|_{\text{BL}_2}^2 \sum_{i=0}^n \sum_{j=0}^n \hat{\beta}_i \hat{\beta}_j \lambda_i^x \lambda_j^y \phi_{2,2}(\|\mathbf{x} - \mathbf{y}\|) \right]^{1/2}, \quad (2.24)$$

for any $g \in \text{BL}_2(\mathbb{R}^2)$, $\mathbf{x} = (x_1, x_2)^T$, $\mathbf{y} = (y_1, y_2)^T$ and $\phi_{2,2}(r) = r^2 \log(r)$, $r \geq 0$.

This lemma enables us to estimate the error at a given point $\tilde{\mathbf{x}}$, if the interpolation data are cell averages.

Theorem 2.9 *Let the triangles T_i , $i = 1, \dots, n$ with vertices $\mathbf{a}_{i1}, \mathbf{a}_{i2}, \mathbf{a}_{i3}$ and centres $\mathbf{a}_{ic} = (\mathbf{a}_{i1} + \mathbf{a}_{i2} + \mathbf{a}_{i3})/3$ be assigned to the functionals (cell average operators) λ_i^x , $i = 1, \dots, n$ defined by*

$$\lambda_i^x(u) := \frac{1}{|T_i|} \int_{T_i} u(\mathbf{x}) \, d\mathbf{x}, \quad i = 1, \dots, n.$$

Let $\lambda_0^x = \delta_{\tilde{\mathbf{x}}}$ be the point evaluation at $\tilde{\mathbf{x}}$ and let $\hat{\beta}_i$, $i = 1, \dots, n$ be given by

$$\hat{\beta}_0 = -1, \quad (2.25)$$

$$\hat{\beta}_i = \beta_i, \quad \beta_i > 0, \quad i = 1, \dots, n, \quad \text{and} \quad \sum_{i=1}^n \beta_i = 1, \quad (2.26)$$

such that

$$\tilde{\mathbf{x}} = \sum_{i=1}^n \beta_i \mathbf{a}_{ic}.$$

Then we obtain

$$|u(\tilde{\mathbf{x}}) - s(\tilde{\mathbf{x}})| \leq [8\pi \|u\|_{\text{BL}_2}^2 \Phi(\beta)]^{1/2} \quad (2.27)$$

for all $u \in \text{BL}_2(\mathbb{R}^2)$, where $\beta = \{\beta_i\}_{i=1}^n$ and Φ is given by

$$\Phi(\beta) = \sum_{i=1}^n \sum_{j=1}^n \beta_i \beta_j \lambda_i^x \lambda_j^y \phi_{2,2}(\|\mathbf{x} - \mathbf{y}\|) - 2 \sum_{i=1}^n \beta_i \lambda_i^y \phi_{2,2}(\|\tilde{\mathbf{x}} - \mathbf{y}\|), \quad (2.28)$$

and s denotes the thin plate spline interpolant with respect to the data $\lambda_i^x(u) = \lambda_i^x(s)$, $i = 1, \dots, n$.

Proof. Let $g = u - s$ so that

$$\hat{L}g = \sum_{i=0}^n \hat{\beta}_i \lambda_i^x g = s(\tilde{\mathbf{x}}) - u(\tilde{\mathbf{x}}).$$

To be able to use the result (2.24) in Lemma 2.8 in the proof of this theorem, we need to make sure that the two conditions on the $\hat{\beta}_i$'s in (2.23) are satisfied. Clearly, with our choices of $\hat{\beta}_i$, $i = 0, 1, \dots, n$ in (2.25) and (2.26), the first condition is satisfied.

To show that the second condition is satisfied, we need to evaluate

$$\sum_{i=0}^n \hat{\beta}_i \lambda_i^x \mathbf{x} = \sum_{i=0}^n \hat{\beta}_i \lambda_i^x \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

We do this by mapping each triangle T_i with vertices $\mathbf{a}_{i1} = (x_{1i}^1, x_{2i}^1)$, $\mathbf{a}_{i2} = (x_{1i}^2, x_{2i}^2)$, $\mathbf{a}_{i3} = (x_{1i}^3, x_{2i}^3)$ to a canonical reference triangle K with vertices $\hat{\mathbf{a}}_1 = (0, 0)$, $\hat{\mathbf{a}}_2 = (1, 0)$, $\hat{\mathbf{a}}_3 = (0, 1)$ by a unique invertible affine mapping F_i such that

$$\mathbf{x} = F_i(\mathbf{v}) = B_i \mathbf{v} + \mathbf{a}_{i1}, \quad (2.29)$$

where $\mathbf{x} = (x_1, x_2) \in T_i$, $\mathbf{v} = (v_1, v_2) \in K$, B_i is an invertible 2×2 matrix and

$$F_i(\hat{\mathbf{a}}_\ell) = \mathbf{a}_{i\ell}, \quad \ell = 1, 2, 3.$$

The matrix B_i is given as

$$B_i = \begin{pmatrix} x_{1i}^2 - x_{1i}^1 & x_{1i}^3 - x_{1i}^1 \\ x_{2i}^2 - x_{2i}^1 & x_{2i}^3 - x_{2i}^1 \end{pmatrix}. \quad (2.30)$$

Hence, we have the relations

$$\begin{aligned} x_1 &= x_{1i}^1 + (x_{1i}^2 - x_{1i}^1)v_1 + (x_{1i}^3 - x_{1i}^1)v_2, \\ x_2 &= x_{2i}^1 + (x_{2i}^2 - x_{2i}^1)v_1 + (x_{2i}^3 - x_{2i}^1)v_2. \end{aligned}$$

If we invert this relationship, we find that

$$\begin{aligned} v_1 &= \frac{(x_1 - x_{1i}^1)(x_{2i}^3 - x_{2i}^1) - (x_2 - x_{2i}^1)(x_{1i}^3 - x_{1i}^1)}{J_i} \\ v_2 &= \frac{(x_2 - x_{2i}^1)(x_{1i}^2 - x_{1i}^1) - (x_1 - x_{1i}^1)(x_{2i}^2 - x_{2i}^1)}{J_i}, \end{aligned}$$

where the Jacobian J_i of the mapping is given by

$$J_i = \det(B_i).$$

Now, $|T_i| = J_i|K|$, $|K| = \frac{1}{2}$ and $dx_1 dx_2 = J_i dv_1 dv_2$; therefore,

$$\int_{T_i} x_1 dx_1 dx_2 = \frac{1}{6} J_i (x_{1i}^1 + x_{1i}^2 + x_{1i}^3) \quad \text{and} \quad \int_{T_i} x_2 dx_1 dx_2 = \frac{1}{6} J_i (x_{2i}^1 + x_{2i}^2 + x_{2i}^3).$$

All this means that

$$\begin{aligned} \sum_{i=0}^n \hat{\beta}_i \lambda_i^x \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= -\tilde{\mathbf{x}} + \sum_{i=1}^n \frac{\beta_i}{|T_i|} \begin{pmatrix} \frac{1}{6} J_i (x_{1i}^1 + x_{1i}^2 + x_{1i}^3) \\ \frac{1}{6} J_i (x_{2i}^1 + x_{2i}^2 + x_{2i}^3) \end{pmatrix} \\ &= -\tilde{\mathbf{x}} + \sum_{i=1}^n \frac{\beta_i}{J_i |K|} \frac{1}{2} J_i \mathbf{a}_{ic} \\ &= -\tilde{\mathbf{x}} + \sum_{i=1}^n \beta_i \mathbf{a}_{ic} \\ &= 0, \end{aligned} \tag{2.31}$$

showing that the second condition is also satisfied. We then conclude by Lemma 2.8 that

$$|u(\tilde{\mathbf{x}}) - s(\tilde{\mathbf{x}})| \leq [8\pi \|g\|_{\text{BL}_2}^2 \Phi(\beta)]^{1/2}. \tag{2.32}$$

Since the interpolant s minimises the energy $|\cdot|_{\text{BL}_2(\mathbb{R}^2)}$ among all interpolants $f \in \text{BL}_2(\mathbb{R}^2)$ satisfying

$$\lambda_i^x f = \lambda_i^x u, \quad i = 1, \dots, n$$

we obtain

$$\begin{aligned} \|g\|_{\text{BL}_2(\mathbb{R}^2)}^2 &= \|u - s\|_{\text{BL}_2(\mathbb{R}^2)}^2 = (u - s, u - s)_{\text{BL}_2(\mathbb{R}^2)} \\ &= (u, u)_{\text{BL}_2(\mathbb{R}^2)} - (u, s)_{\text{BL}_2(\mathbb{R}^2)} - 2(s, u - s)_{\text{BL}_2(\mathbb{R}^2)} + (s, u - s)_{\text{BL}_2(\mathbb{R}^2)} \\ &= (u, u)_{\text{BL}_2(\mathbb{R}^2)} - 2(s, u - s)_{\text{BL}_2(\mathbb{R}^2)} - (s, s)_{\text{BL}_2(\mathbb{R}^2)} \\ &= \|u\|_{\text{BL}_2(\mathbb{R}^2)}^2 - \underbrace{2(s, u - s)_{\text{BL}_2(\mathbb{R}^2)}}_{=0} - \|s\|_{\text{BL}_2(\mathbb{R}^2)}^2 \\ &\leq \|u\|_{\text{BL}_2(\mathbb{R}^2)}^2. \end{aligned} \tag{2.33}$$

This concludes the proof. \square

A more precise form of the error bound (2.27) can be obtained by finding an estimate of the quadratic form $\Phi(\beta)$. However, it is not clear to us at the moment how to obtain this estimate for unstructured triangular meshes.

Fortunately, Wendland [143] provides general convergence results for reconstruction processes from cell averages using conditionally positive definite functions. We present a summary of his results concerning polyharmonic splines below.

Theorem 2.10 *Suppose Ω is bounded and satisfies an interior cone condition. Suppose further $k > d/2$ and $1 \leq q \leq \infty$. If Ω is covered by volumes $\{V_j\}$ such that every ball $B \subseteq \Omega$ of radius h contains at least one volume V_j . Then, the error between $u \in W_k^2$ and its optimal recovery s from cell averages using the polyharmonic spline $\phi_{d,k}$ has the error estimate*

$$\|u - s\|_{L_q(\Omega)} \leq Ch^{k-d(1/2-1/q)_+} |u|_{\text{BL}_k(\Omega)},$$

where $(x)_+ = \max\{x, 0\}$.

Proof. See Wendland [143], Theorem 5.2 and Corollary 6.1. □

For the case where $q = \infty$, this yields

$$\|u - s\|_{L_\infty(\Omega)} \leq Ch^{k-d/2} |u|_{\text{BL}_k(\Omega)}.$$

Hence, when $k = d = 2$, using the thin plate splines leads to a first order scheme. He further showed that under additional assumptions on the function u , improved error estimates can be obtained.

Theorem 2.11 ([143]) *Under the assumptions of Theorem 2.10, we assume that $u \in W_2^{2k}(\Omega)$ has support in Ω . Then the error between u and its optimal recovery s can be bounded by*

$$\|u - s\|_{L_q(\Omega)} \leq Ch^{2k-d(1/2-1/q)_+} \|\Delta^k u\|_{L_2(\Omega)}.$$

Proof. See Wendland [143], Theorem 6.2. □

Furthermore, for the conditionally positive define function $\phi(r) = r$ in \mathbb{R}^2 which we use in Chapters 3 and 6, we note that there is a general result in [144] on optimal recovery from cell averages with conditionally positive definite functions but no specific result for $\phi(r) = r$ in \mathbb{R}^2 . However, in [36, 143], there is a result on interpolation with $\phi(r) = r^\beta$, $\beta \neq 2\mathbb{N}$. It asserts that there exists constants h_0, C such that

$$|u - s| \leq Ch^{\beta/2} |u|_{\mathcal{N}_\phi},$$

provided $h \leq h_0$.

We will denote $\phi(r) = r$ as $\phi_1(r)$ in the rest of this work.

All the finite volume methods that will be designed and implemented in this work

are based on local reconstruction methods. To this end, the next section is concerned with the analysis of local reconstruction by polyharmonic splines which is based on a scaled interpolation problem. This formulation allows us to construct a numerically stable algorithm for the evaluation of polyharmonic spline interpolants.

2.3 Generalized Local Interpolation by Polyharmonic Splines

As regards the discussion in this section, for some fixed point $\mathbf{x}_0 \in \mathbb{R}^d$ and any $h > 0$, we seek to solve the *scaled* interpolation problem

$$\lambda_j^x s^h(\mathbf{x}_0 + h\mathbf{x}) = \lambda_j^x u(\mathbf{x}_0 + h\mathbf{x}), \quad 1 \leq j \leq n, \quad (2.34)$$

where $\Xi = \{\lambda_1^x, \dots, \lambda_n^x\}$ is a \mathcal{P}_k^d -unisolvent set of functionals which we take to be cell average operators in \mathbb{R}^d . If we let $\mathbf{x}_0 = 0$, then the unique generalized polyharmonic spline interpolant s^h is of the form

$$s^h(h\mathbf{x}) = \sum_{j=1}^n c_j^h \lambda_j^y \phi_{d,k}(\|h\mathbf{x} - h\mathbf{y}\|) + p(h\mathbf{x}), \quad p \in \mathcal{P}_k^d, \quad (2.35)$$

satisfying (2.34) and the coefficients c_1^h, \dots, c_n^h satisfy the constraints

$$\sum_{j=1}^n c_j^h \lambda_j^x p(h\mathbf{x}) = 0, \quad \text{for all } p \in \mathcal{P}_k^d. \quad (2.36)$$

The coefficients of the interpolant s^h in (2.35) are obtained by solving the linear system

$$\underbrace{\begin{bmatrix} A_h & P_h \\ P_h^T & \mathbf{0} \end{bmatrix}}_{\mathbf{A}_h} \underbrace{\begin{bmatrix} \mathbf{c}^h \\ \mathbf{d}^h \end{bmatrix}}_{\mathbf{b}_h} = \underbrace{\begin{bmatrix} u|_{h\Xi} \\ 0 \end{bmatrix}}_{u_h}, \quad (2.37)$$

where $A_h = (\lambda_i^x \lambda_j^y (\phi_{d,k}(\|h\mathbf{x} - h\mathbf{y}\|)))_{1 \leq i, j \leq n} \in \mathbb{R}^{n \times n}$, $P_h = (\lambda_j^x (\mathbf{x})^\alpha)_{1 \leq j \leq n; |\alpha| < k} \in \mathbb{R}^{n \times q}$, $u|_{h\Xi} = (\lambda_1^x u(h\mathbf{x}), \dots, \lambda_n^x u(h\mathbf{x}))^T$, $\mathbf{c}^h = (c_1^h, \dots, c_n^h)^T$, and $\mathbf{d}^h = (d_\alpha^h)_{|\alpha| < k} \in \mathbb{R}^q$ for the coefficients of the polynomial part in (2.35).

The Lagrange-type representation of s^h in (2.35) is given by

$$s^h(h\mathbf{x}) = \sum_{j=1}^n \ell_j^h(h\mathbf{x}) \lambda_j^x u(h\mathbf{x}) \quad (2.38)$$

where

$$p(h\mathbf{x}) = \sum_{j=1}^n \ell_j^h(h\mathbf{x}) \lambda_j^x p(h\mathbf{x}) \quad \text{for all } p \in \mathcal{P}_k^d, \quad (2.39)$$

and the *cardinal functions* $\ell_1^h(h\mathbf{x}), \dots, \ell_n^h(h\mathbf{x})$ satisfy

$$\lambda_j^x(\ell_i^h(h\mathbf{x})) = \begin{cases} 1, & \text{for } i = j, \\ 0, & \text{for } i \neq j, \end{cases} \quad 1 \leq i, j \leq n. \quad (2.40)$$

Moreover, for a point $h\mathbf{x} \in \mathbb{R}^d$, the vectors $\ell^h(h\mathbf{x}) = (\ell_1^h(h\mathbf{x}), \dots, \ell_n^h(h\mathbf{x}))^T$ and $v^h(h\mathbf{x}) = (v_1(h\mathbf{x}), \dots, v_q(h\mathbf{x}))^T$ are the unique solution of the linear system

$$\underbrace{\begin{bmatrix} A_h & P_h \\ P_h^T & \mathbf{0} \end{bmatrix}}_{\mathbf{A}_h} \underbrace{\begin{bmatrix} \ell^h(h\mathbf{x}) \\ v^h(h\mathbf{x}) \end{bmatrix}}_{\nu^h(h\mathbf{x})} = \underbrace{\begin{bmatrix} R_h(h\mathbf{x}) \\ S_h(h\mathbf{x}) \end{bmatrix}}_{\beta_h(h\mathbf{x})}, \quad (2.41)$$

where $R_h(h\mathbf{x}) = (\lambda_j^y \phi_{d,k}(\|h\mathbf{x} - h\mathbf{y}\|))_{1 \leq j \leq n}$ and $S_h(h\mathbf{x}) = ((h\mathbf{x})^\alpha)_{|\alpha| < k} \in \mathbb{R}^q$.

We will now show that this Lagrange-type representation does exist. Following [117] and using the vectors $R_h(h\mathbf{x})$ and $S_h(h\mathbf{x})$, the polyharmonic spline interpolant (2.35) can be re-written in the form

$$\begin{aligned} s^h(h\mathbf{x}) &= (\mathbf{c}^h)^T R_h(h\mathbf{x}) + (\mathbf{d}^h)^T S_h(h\mathbf{x}) = ((\mathbf{c}^h)^T, (\mathbf{d}^h)^T) \begin{pmatrix} R_h(h\mathbf{x}) \\ S_h(h\mathbf{x}) \end{pmatrix} \\ &= ((\mathbf{c}^h)^T, (\mathbf{d}^h)^T) \mathbf{A}_h \begin{pmatrix} \ell^h(h\mathbf{x}) \\ v^h(h\mathbf{x}) \end{pmatrix} \\ &= ((\mathbf{c}^h)^T A_h + (\mathbf{d}^h)^T P_h^T) \ell^h(h\mathbf{x}) + (\mathbf{c}^h)^T P_h v^h(h\mathbf{x}). \end{aligned}$$

Using (2.37) and due to the symmetry A_h ,

$$A_h \mathbf{c}^h + P_h \mathbf{d}^h = [(\mathbf{c}^h)^T A_h + (\mathbf{d}^h)^T P_h^T]^T = u|_{h\Xi},$$

and

$$P_h^T \mathbf{c}^h = [(\mathbf{c}^h)^T P_h]^T = 0.$$

Therefore,

$$s^h(h\mathbf{x}) = (u|_{h\Xi})^T \ell^h(h\mathbf{x}) = \sum_{i=1}^n \ell_i^h(h\mathbf{x}) \lambda_i^x u(h\mathbf{x}).$$

In conclusion, starting with the Lagrange representation of s^h in (2.38), we obtain

$$\begin{aligned} s^h(h\mathbf{x}) &= \langle \ell^h(h\mathbf{x}), u|_{h\Xi} \rangle = \langle \nu^h(h\mathbf{x}), u_h \rangle \\ &= \langle \mathbf{A}_h^{-1} \beta_h(h\mathbf{x}), u_h \rangle = \langle \beta_h(h\mathbf{x}), \mathbf{A}_h^{-1} u_h \rangle \\ &= \langle \beta_h(h\mathbf{x}), \mathbf{b}_h \rangle \end{aligned}$$

where u_h is defined in (2.37). This expression uses the two representations of s^h in (2.35) and (2.38).

2.3.1 Local approximation order and numerical stability

Definition 2.12 *Let $n \in \mathbb{N}$ be a fixed number of \mathcal{P}_k^d -unisolvent functionals λ_i , $i = 1, \dots, n$, which are independent of h and let s^h denote the polyharmonic spline interpolant satisfying (2.34). We say that the approximation order of the local polyharmonic spline interpolation at $\mathbf{x}_0 \in \mathbb{R}^d$ with respect to the function space \mathcal{F} is p , iff for $f \in \mathcal{F}$ the asymptotic bound*

$$|u(\mathbf{x}_0 + h\mathbf{x}) - s^h(\mathbf{x}_0 + h\mathbf{x})| = \mathcal{O}(h^p), \quad h \rightarrow 0, \quad (2.42)$$

holds for any $\mathbf{x} \in \mathbb{R}^d$.

We now state and prove an important lemma which will be used in our subsequent discussions.

Lemma 2.13 ([63]) *For any $h > 0$, let $\ell^h(h\mathbf{x})$ be the solution in (2.41). Then,*

$$\ell^h(h\mathbf{x}) = \ell^1(\mathbf{x}), \quad \text{for every } \mathbf{x} \in \mathbb{R}^d. \quad (2.43)$$

The proof we present below is completely analogous to the one presented in [63]. It is modified here for the case of generalized interpolation with cell average operators rather than the point evaluations considered in [63].

Proof. Let

$$\mathcal{S}_h = \left\{ \sum_{j=1}^n c_j^h \lambda_j^y \phi_{d,k}(\|\cdot - h\mathbf{y}\|) + p : p \in \mathcal{P}_k^d, \quad \sum_{j=1}^n c_j^h \lambda_j^x q(\mathbf{x}) = 0 \text{ for all } q \in \mathcal{P}_k^d \right\}$$

be the space of all possible generalized polyharmonic spline interpolants of the form (2.35) satisfying (2.34) for a \mathcal{P}_k^d -unisolvent set of functionals $\Xi = \{\lambda_1^x, \dots, \lambda_n^x\}$.

We need to show that \mathcal{S}_h is a scaled version of \mathcal{S}_1 , so that

$$\mathcal{S}_h = \{\sigma_h(s) : s \in \mathcal{S}_1\} \quad (2.44)$$

where we define the dilation operator as $\sigma_h(s) = s(\cdot/h)$, $h > 0$. Thus, due to the unicity of the interpolation in \mathcal{S}_h or \mathcal{S}_1 , their Lagrange basis functions must coincide by satisfying $\ell^h = \sigma_h(\ell^1)$. Therefore, we need to show that $\mathcal{S}_h = \sigma_h(\mathcal{S}_1)$.

When d is odd, $\mathcal{S}_h = \sigma_h(\mathcal{S}_1)$ follows from the homogeneity of $\phi_{d,k}$, where

$$\phi_{d,k}(hr) = h^{2k-d}\phi_{d,k}(r).$$

When d is even,

$$\phi_{d,k}(hr) = h^{2k-d}(\phi_{d,k}(r) + r^{2k-d}\log(h)),$$

and so any function $s^h \in \mathcal{S}_h$ has, for some $p \in \mathcal{P}_k^d$, the form

$$\begin{aligned} s^h(h\mathbf{x}) &= \sum_{j=1}^n c_j^h \lambda_j^y \phi_{d,k}(\|h\mathbf{x} - h\mathbf{y}\|) + p(\mathbf{x}), \\ &= \sum_{j=1}^n c_j^h \lambda_j^y \{h^{2k-d}\phi_{d,k}(\|\mathbf{x} - \mathbf{y}\|) + h^{2k-d}\|\mathbf{x} - \mathbf{y}\|^{2k-d}\log(h)\} + p(\mathbf{x}), \\ &= h^{2k-d} \left(\sum_{j=1}^n c_j^h \lambda_j^y \phi_{d,k}(\|\mathbf{x} - \mathbf{y}\|) + \log(h)g(\mathbf{x}) \right) + p(\mathbf{x}), \end{aligned}$$

where

$$g(\mathbf{x}) = \sum_{j=1}^n c_j^h \lambda_j^y \|\mathbf{x} - \mathbf{y}\|^{2k-d}.$$

To establish that s^h is in $\sigma_h(\mathcal{S}_1)$, we need to show that the g is a polynomial of degree at most $k-1$. We therefore write g as

$$\begin{aligned} g(\mathbf{x}) &= \sum_{j=1}^n c_j^h \lambda_j^y \left(\sum_{|\alpha|+|\beta|=2k-d} c_{\alpha,\beta} \cdot \mathbf{x}^\alpha \mathbf{y}^\beta \right), \\ &= \sum_{|\alpha|+|\beta|=2k-d} c_{\alpha,\beta} \cdot \mathbf{x}^\alpha \sum_{j=1}^n c_j^h \lambda_j^y \mathbf{y}^\beta, \end{aligned}$$

for some coefficients $c_{\alpha,\beta} \in \mathbb{R}$ with $|\alpha| + |\beta| = 2k - d$. Now due to the vanishing moment conditions

$$\sum_{j=1}^n c_j^h \lambda_j^x p(h\mathbf{x}) = 0, \quad \text{for all } p \in \mathcal{P}_k^d$$

for the coefficients c_1^h, \dots, c_n^h , this means that the degree of g is at most $2k - d - k =$

$k - d < k$. Therefore, $s^h \in \sigma_h(\mathcal{S}_1)$, and so $\mathcal{S}_h \subset \sigma_h(\mathcal{S}_1)$. Similarly, $\mathcal{S}_1 \subset \sigma_h^{-1}(\mathcal{S}_h)$. We then conclude that $\mathcal{S}_h = \sigma_h(\mathcal{S}_1)$ for any d and this completes the proof. \square

The following theorem summarizes the result on local approximation order.

Theorem 2.14 *Let $u \in C^k$ in a region containing \mathbf{x}_0 . Then the local approximation order of polyharmonic splines $\phi_{d,k}$ is k , i.e.*

$$|u(\mathbf{x}_0 + h\mathbf{x}) - s^h(\mathbf{x}_0 + h\mathbf{x})| = \mathcal{O}(h^k), \quad h \rightarrow 0 \quad (2.45)$$

where s^h denotes the polyharmonic spline interpolant satisfying (2.34).

Proof. We assume $\mathbf{x}_0 = 0$ without loss of generality and we use the representation (2.38) for s^h . For any $u \in C^k$, any $\mathbf{x} \in \mathbb{R}^d$, and $h > 0$, we define the k -th order Taylor polynomial

$$T_k(\mathbf{y}) = \sum_{|\alpha| < k} \frac{1}{\alpha!} D^\alpha u(h\mathbf{x})(\mathbf{y} - h\mathbf{x})^\alpha \quad (2.46)$$

of u around $h\mathbf{x}$. This then means that

$$u(h\mathbf{x}) = T_k(\mathbf{y}) - \sum_{0 < |\alpha| < k} \frac{1}{\alpha!} D^\alpha u(h\mathbf{x})(\mathbf{y} - h\mathbf{x})^\alpha$$

and thus by the polynomial reproduction property (2.39) we have

$$u(h\mathbf{x}) = T_k(h\mathbf{x}) = \sum_{j=1}^n \ell_j^h(h\mathbf{x}) \lambda_j^x(T_k(h\mathbf{x})). \quad (2.47)$$

From (2.38) and (2.47) we obtain

$$u(h\mathbf{x}) - s^h(h\mathbf{x}) = \sum_{j=1}^n \ell_j^h(h\mathbf{x}) [\lambda_j^x(T_k(h\mathbf{x})) - \lambda_j^x(u(h\mathbf{x}))]. \quad (2.48)$$

Due to Lemma 2.13, the Lebesgue constant

$$\Lambda = \sup_{h>0} \sum_{j=1}^n |\ell_j^h(h\mathbf{x})| = \sum_{j=1}^n |\ell_j^1(\mathbf{x})|, \quad (2.49)$$

is bounded locally around the origin $\mathbf{x}_0 = 0$. We conclude that

$$|u(h\mathbf{x}) - s^h(h\mathbf{x})| = \mathcal{O}(h^k), \quad h \rightarrow 0.$$

\square

Remark 2.15 *When we use local reconstruction, we observe that the local approximation order of the polyharmonic spline reconstruction method is arbitrarily high. More precisely, when working with $\phi \equiv \phi_{d,k}$ the local approximation order is k , and so the smoothness parameter k in $\phi_{d,k}$ can be used to obtain a desired target approximation order k .*

It is well known that the stability of an interpolation scheme depends on the conditioning of the given problem. This is a key issue in the design and implementation of any useful interpolation or reconstruction scheme. To discuss the conditioning of the reconstruction by polyharmonic splines, suppose $\Omega \in \mathbb{R}^d$ is a finite computational domain and $\Xi = \{\lambda_1^x, \dots, \lambda_n^x\}$ is a \mathcal{P}_k^d -unisolvent set of functionals. The interpolation operator $\mathcal{R}_{d,k} : C(\Omega) \mapsto C(\Omega)$, yields for any function $u \in C(\Omega)$ the polyharmonic spline recovery function $\mathcal{R}_{d,k}u = s \in C(\Omega)$ of the form

$$s(\mathbf{x}) = \sum_{j=1}^n c_j \lambda_j^y \phi_{d,k}(\|\mathbf{x} - \mathbf{y}\|) + p(\mathbf{x}), \quad p \in \mathcal{P}_k^d, \quad (2.50)$$

satisfying $\lambda_i^x(s) = \lambda_i^x(u)$, $i = 1, \dots, n$.

Definition 2.16 *The condition number of an interpolation operator $\mathcal{R} : C(\Omega) \mapsto C(\Omega)$, $\Omega \in \mathbb{R}^d$ with respect to the L_∞ -norm is the smallest number κ_∞ satisfying*

$$\|\mathcal{R}u\|_{L_\infty(\Omega)} \leq \kappa_\infty \cdot \|u\|_{L_\infty(\Omega)} \text{ for all } u \in C(\Omega).$$

Moreover, κ_∞ is the operator norm of \mathcal{R} with respect to the L_∞ -norm.

The following results in [64] are necessary for the discussion on the stable evaluation of polyharmonic splines.

Theorem 2.17 *The condition number κ_∞ of interpolation by polyharmonic spline is given by the Lebesgue constant*

$$\Lambda(\Omega, \Xi) = \max_{\mathbf{x} \in \Omega} \sum_{j=1}^n |\ell_j(\mathbf{x})|. \quad (2.51)$$

Moreover, Lemma 2.13 and Theorem 2.17, yield the following result on the stability of interpolation by polyharmonic splines.

Theorem 2.18 *The absolute condition number of polyharmonic spline interpolation is invariant under rotations, translations and uniform scalings.*

Theorem 2.18 implies that the conditioning of the interpolation scheme depends on the geometry of the cells assigned to the functionals λ_i^x with respect to the center \mathbf{x}_0 , but not on the scale h . But since the spectral condition number of the matrix A_h in (2.37) depends on h , a simple re-scaling can be implemented as a way of preconditioning the matrix A_h for very small h , see [63, 64]. To this end, we evaluate the polyharmonic spline interpolant s^h as follows

$$\begin{aligned} s^h(h\mathbf{x}) &= \langle \ell^h(h\mathbf{x}), u|_{h\Xi} \rangle = \langle \ell^1(\mathbf{x}), u|_{h\Xi} \rangle \\ &= \langle \nu^1(\mathbf{x}), u_h \rangle = \langle \mathbf{A}_1^{-1} \beta_1(\mathbf{x}), u_h \rangle \\ &= \langle \beta_1(\mathbf{x}), \mathbf{A}_1^{-1} u_h \rangle \end{aligned} \quad (2.52)$$

where $u|_{h\Xi} = (\lambda_1^x(u(h\mathbf{x})), \dots, \lambda_n^x(u(h\mathbf{x})))^T$ and the last expression in (2.52) is the stable form we prefer to work with. From (2.52), we can evaluate s^h at $h\mathbf{x}$ by solving the linear system

$$\mathbf{A}_1 \alpha = u_h. \quad (2.53)$$

The solution $\alpha \in \mathbb{R}^{n+q}$ in (2.53) then yields the coefficients of $s^h(h\mathbf{x})$ with respect to the basis functions in $\beta_1(\mathbf{x})$.

2.3.2 Derivatives of polyharmonic splines

Our motivation for analyzing the computation, approximation order and stable evaluation of derivatives of local polyharmonic spline interpolant comes from their application in the construction of the ADER-WENO schemes which are the subject of Chapter 4. A recovery function and its derivatives are used for the initial data of the Generalized Riemann Problem which is the basis of the high order flux evaluation of the ADER-WENO method. Unlike previous ADER-WENO methods that rely on polynomial reconstruction methods, the ADER-WENO method in this work uses a WENO reconstruction based on polyharmonic splines for its spatial discretisation. We note that the derivatives of polyharmonic splines are not as straightforward to compute as those of polynomials and for the sake of numerical stability care must be taken in evaluating them.

Now, suppose we have the polyharmonic spline interpolant

$$s(\mathbf{x}) = \sum_{j=1}^n c_j \lambda_j^y \phi_{d,k}(\|\mathbf{x} - \mathbf{y}\|) + p(\mathbf{x}), \quad p \in \mathcal{P}_k^d, \quad (2.54)$$

then

$$D^\gamma s(\mathbf{x}) = \sum_{j=1}^n c_j \lambda_j^y D^\gamma \phi_{d,k}(\|\mathbf{x} - \mathbf{y}\|) + D^\gamma p(\mathbf{x}), \quad p \in \mathcal{P}_k^d. \quad (2.55)$$

We note that for $\mathbf{x} = (x_1, \dots, x_d)^T \in \mathbb{R}^d$ and $\gamma = (\gamma_1, \dots, \gamma_d)^T \in \mathbb{N}^d$

$$D^\gamma := \left(\frac{\partial}{\partial x_1} \right)^{\gamma_1} \cdots \left(\frac{\partial}{\partial x_d} \right)^{\gamma_d},$$

where $\frac{\partial}{\partial x_i}$ denotes the partial derivative with respect to x_i , $i = 1, \dots, d$ and $|\gamma| = \gamma_1 + \dots + \gamma_d$. D^γ is the identity operator when $\gamma = 0$. Alternatively, if we use the Lagrange-type representation

$$s(\mathbf{x}) = \sum_{j=1}^n \ell_j(\mathbf{x}) \lambda_j^x u(\mathbf{x}), \quad (2.56)$$

then

$$D^\gamma s(\mathbf{x}) = \sum_{j=1}^n D^\gamma \ell_j(\mathbf{x}) \lambda_j^x u(\mathbf{x}), \quad (2.57)$$

where the vectors $D^\gamma \ell(\mathbf{x}) = (D^\gamma \ell_1(\mathbf{x}), \dots, D^\gamma \ell_n(\mathbf{x}))^T$ and $D^\gamma v(\mathbf{x}) = (D^\gamma v_1(\mathbf{x}), \dots, D^\gamma v_q(\mathbf{x}))^T$ are the unique solution of the linear system

$$\begin{bmatrix} A & P \\ P^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} D^\gamma \ell(\mathbf{x}) \\ D^\gamma v(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} D^\gamma R(\mathbf{x}) \\ D^\gamma S(\mathbf{x}) \end{bmatrix}, \quad (2.58)$$

where

$$D^\gamma R(\mathbf{x}) = (\lambda_j^y D^\gamma \phi_{d,k}(\|\mathbf{x} - \mathbf{y}\|))_{1 \leq j \leq n} \in \mathbb{R}^n, \quad \text{and} \quad D^\gamma S(\mathbf{x}) = (D^\gamma(\mathbf{x}^\alpha))_{|\alpha| < k} \in \mathbb{R}^q. \quad (2.59)$$

The system (2.58) is obtained by the formal differentiation of (2.11) [144].

It is clear from (2.55) and (2.59) that in computing the derivative of $s(\mathbf{x})$, we first need to compute $D^\gamma \phi_{d,k}(\|\mathbf{x} - \mathbf{y}\|)$. Computing the derivatives of the polynomial part $p(\mathbf{x})$ or $S(\mathbf{x})$ is trivial.

Derivatives of the radial function $\phi_{d,k}(\|\mathbf{x} - \mathbf{y}\|)$

For the sake of the numerical examples in Chapter 4, we first of all write out clearly the following derivatives for $d = 2$. Let

$$\Phi_{2,k}(\mathbf{x} - \mathbf{y}) = \phi_{2,k}(\|\mathbf{x} - \mathbf{y}\|) = \|\mathbf{x} - \mathbf{y}\|^{2k-2} \log \|\mathbf{x} - \mathbf{y}\|,$$

where $\mathbf{x} = (x_1, x_2)^T$, $\mathbf{y} = (y_1, y_2)^T$ and for notational simplicity let $\Delta = \|\mathbf{x} - \mathbf{y}\|$. The first derivatives (for $k \geq 2$) of $\Phi_{2,k}$ are given by

$$\begin{aligned} \frac{\partial \Phi_{2,k}}{\partial \tau} &= (2k-2)(\tau-\theta)\Delta^{2k-4} \log \Delta + (\tau-\theta)\Delta^{2k-4} \\ &= ((\tau-\theta) + (2k-2)(\tau-\theta) \log \Delta) \Delta^{2k-4} \end{aligned}$$

where when $\tau = x_1, \theta = y_1$ and when $\tau = x_2, \theta = y_2$. The second derivatives (for $k \geq 3$) are given by

$$\begin{aligned} \frac{\partial^2 \Phi_{2,k}}{\partial \tau_1 \partial \tau_2} &= (2k-2)(2k-4)(\tau_1-\theta_1)(\tau_2-\theta_2)\Delta^{2k-6} \log \Delta \\ &\quad + 2(2k-3)(\tau_1-\theta_1)(\tau_2-\theta_2)\Delta^{2k-6} + \delta(2k-2)\Delta^{2k-4} \log \Delta \\ &\quad + \delta\Delta^{2k-4} \\ &= \left\{ (2(2k-3)(\tau_1-\theta_1)(\tau_2-\theta_2) + \delta\Delta^2) \right. \\ &\quad \left. + ((2k-2)(2k-4)(\tau_1-\theta_1)(\tau_2-\theta_2) + \delta(2k-2)\Delta^2) \log \Delta \right\} \Delta^{2k-6} \end{aligned}$$

where both τ_1 and τ_2 are equal to either x_1 or x_2 . Furthermore, when $\tau_1 = x_1, \theta_1 = y_1$, when $\tau_1 = x_2, \theta_1 = y_2$, when $\tau_2 = x_1, \theta_2 = y_1$, and when $\tau_2 = x_2, \theta_2 = y_2$. In addition,

$$\delta \equiv \delta_{\tau_1 \tau_2} = \begin{cases} 1, & \tau_1 = \tau_2; \\ 0, & \tau_1 \neq \tau_2. \end{cases}$$

We now state and prove the following results in [144] modified here for the derivatives of $\phi_{d,k}(\|\mathbf{x} - \mathbf{y}\|)$.

Lemma 2.19 *Let $\Phi_{d,k}(\mathbf{x} - \mathbf{y}) = \phi_{d,k}(\Delta) = \Delta^{2k-d} \log \Delta$, ($\Delta = \|\mathbf{x} - \mathbf{y}\|$) with $k \in \mathbb{N}$, $d \in 2\mathbb{N}$, $2k > d$, $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$ and $\mathbf{y} = (y_1, \dots, y_d) \in \mathbb{R}^d$. For every $\gamma \in \mathbb{N}_0^d$, there exists homogenous polynomials $q_{\gamma,k} \equiv q_{\gamma,k}(\mathbf{x} - \mathbf{y})$, $r_{\gamma,k} \equiv r_{\gamma,k}(\mathbf{x} - \mathbf{y}) \in \mathcal{P}_{|\gamma|+1}^d$ such that*

$$D^\gamma \Phi_{d,k}(\mathbf{x} - \mathbf{y}) = (q_{\gamma,k} + r_{\gamma,k} \log \Delta) \Delta^{2k-d-2|\gamma|} \quad (2.60)$$

for $\mathbf{x} - \mathbf{y} \neq 0$.

Proof. The proof is by induction about the length of γ . For $|\gamma| = 0$, there is nothing to show. Now assume $|\gamma| > 0$. Without loss of restriction we assume $\gamma_1 \geq 1$. Define $\nu = (\gamma_1 - 1, \dots, \gamma_d)^T$. Then there exists homogenous polynomials $q_{\nu,k}, r_{\nu,k}$ of degree $|\nu|$ such that

$$\begin{aligned} D^\gamma \Phi_{d,k}(\mathbf{x} - \mathbf{y}) &= \frac{\partial}{\partial x_1} D^\nu \Phi_{d,k}(\mathbf{x} - \mathbf{y}) \\ &= \frac{\partial}{\partial x_1} \left\{ (q_{\nu,k} + r_{\nu,k} \log \Delta) \Delta^{2k-d-2|\nu|} \right\}. \end{aligned}$$

Using the product rule and factorising we obtain

$$\begin{aligned}
D^\gamma \Phi_{d,k}(\mathbf{x} - \mathbf{y}) &= \left[\left(\frac{\partial q_{\nu,k}}{\partial x_1} \Delta^2 + q_{\nu,k} \cdot (x_1 - y_1) + 2 \left(k - \frac{d}{2} - |\nu| \right) r_{\nu,k} \cdot (x_1 - y_1) \right) \right. \\
&\quad \left. + \left(\frac{\partial r_{\nu,k}}{\partial x_1} \Delta^2 + 2 \left(k - \frac{d}{2} - |\nu| \right) r_{\nu,k} \cdot (x_1 - y_1) \right) \log \Delta \right] \Delta^{2k-d-2|\gamma|} \\
&= (q_{\gamma,k} + r_{\gamma,k} \log \Delta) \Delta^{2k-d-2|\gamma|}.
\end{aligned}$$

The polynomials $q_{\gamma,k}$ and $r_{\gamma,k}$ are indeed homogenous polynomials of degree $|\gamma|$, because the derivative of a homogenous polynomial of degree l is a homogenous polynomial of degree $l - 1$ and the product of two homogenous polynomials of degree l and k is a homogenous polynomial of degree $l + k$. \square

Lemma 2.20 *Let $\Phi(\mathbf{x} - \mathbf{y}) = \phi(\Delta) = \Delta^{2k-d}$ ($\Delta = \|\mathbf{x} - \mathbf{y}\|$) with $k, d \in \mathbb{N}$, $2k > d$, $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$ and $\mathbf{y} = (y_1, \dots, y_d) \in \mathbb{R}^d$. For every $\gamma \in \mathbb{N}_0^d$, there exists a homogenous polynomials $\tilde{r}_{\gamma,k} \equiv \tilde{r}_{\gamma,k}(\mathbf{x} - \mathbf{y}) \in \mathcal{P}_{|\gamma|+1}^d$ such that*

$$D^\gamma \Phi(\mathbf{x} - \mathbf{y}) = \tilde{r}_{\gamma,k} \Delta^{2k-d-2|\gamma|} \quad (2.61)$$

for $\mathbf{x} - \mathbf{y} \neq 0$.

Proof. Once again the proof is by induction about the length of γ . For $|\gamma| = 0$, there is nothing to show. Now assume $|\gamma| > 0$. Without loss of restriction we assume $\gamma_1 \geq 1$. Define $\nu = (\gamma_1 - 1, \dots, \gamma_d)^T$. Then there exists homogenous polynomials $\tilde{r}_{\nu,k}$ of degree $|\nu|$ such that

$$\begin{aligned}
D^\gamma \Phi(\mathbf{x} - \mathbf{y}) &= \frac{\partial}{\partial x_1} D^\nu \Phi(\mathbf{x} - \mathbf{y}) \\
&= \frac{\partial}{\partial x_1} \{ \tilde{r}_{\nu,k} \Delta^{2k-d-2|\nu|} \} \\
&= \left(\frac{\partial \tilde{r}_{\nu,k}}{\partial x_1} \Delta^2 + (2k - d - 2|\nu|) \tilde{r}_{\nu,k} \cdot (x_1 - y_1) \right) \Delta^{2k-d-2|\nu|-2} \\
&= \tilde{r}_{\gamma,k} \Delta^{2k-d-2|\gamma|}.
\end{aligned}$$

As in the proof of Lemma 2.19, the polynomial $\tilde{r}_{\gamma,k}$ is a homogenous polynomial of degree $|\gamma|$. \square

Remark 2.21 *If d is even, then $\tilde{r}_{\gamma,k} \equiv r_{\gamma,k}$ with $r_{\gamma,k}$ defined in (2.60).*

The discussion that follows is based on local generalized interpolation using the scaled interpolation problem (2.34) and the unique interpolant of the form (2.35) and (2.38).

We first of all state and prove a lemma concerning the derivatives of the Lagrange basis function which will serve as a basis for our discussion on the stable evaluation of derivatives of polyharmonic splines and local approximation order.

Lemma 2.22 *For any $h > 0$, let $\ell^h(h\mathbf{x})$ be the solution in (2.41). Then,*

$$D^\gamma \ell^h(h\mathbf{x}) = h^{-|\gamma|} D^\gamma \ell^1(\mathbf{x}) \quad \text{for every } \mathbf{x} \in \mathbb{R}^d \text{ and } |\gamma| < k.$$

Proof. We use the form (2.38) of the interpolant in this proof. The evaluation of each $\ell_i^h(h\mathbf{x})$, $i = 1, \dots, n$ at $h\mathbf{x}$ in (2.38) is given by the solution of the linear system (2.41). Moreover, the derivative of s^h in (2.38) is given

$$D^\gamma s^h(h\mathbf{x}) = \sum_{i=1}^n D^\gamma \ell_i^h(h\mathbf{x}) \lambda_i^x u(h\mathbf{x}) \quad (2.62)$$

where $D^\gamma \ell^h(h\mathbf{x})$ and $D^\gamma v^h(h\mathbf{x})$ are the unique solution of the linear system

$$\underbrace{\begin{bmatrix} A_h & P_h \\ P_h^T & 0 \end{bmatrix}}_{\mathbf{A}_h} \underbrace{\begin{bmatrix} D^\gamma \ell^h(h\mathbf{x}) \\ D^\gamma v^h(h\mathbf{x}) \end{bmatrix}}_{D^\gamma v^h(h\mathbf{x})} = \underbrace{\begin{bmatrix} D^\gamma R_h(h\mathbf{x}) \\ D^\gamma S_h(h\mathbf{x}) \end{bmatrix}}_{D^\gamma \beta_h(h\mathbf{x})}. \quad (2.63)$$

For the purpose of this proof, we write $S_h(h\mathbf{x})$ in the form $S_h(h\mathbf{x}) = (p_1(h\mathbf{x}), \dots, p_q(h\mathbf{x}))^T$, $q = \dim(\mathcal{P}_k^d)$. We let $h = 1$ for the moment and expand for an index $j = 1, \dots, n$, the j -th row of the system (2.63).

When d is odd, this gives

$$\sum_{j=1}^n (D^\gamma \ell_j^1(\mathbf{x})) \lambda_i^x \lambda_j^y \Delta^{2k-d} + \sum_{l=1}^q (D^\gamma v_l^1(\mathbf{x})) \lambda_i^x p_l(\mathbf{x}) = \lambda_i^y (\tilde{r}_{\gamma,k} \Delta^{2k-d-2|\gamma|}). \quad (2.64)$$

Multiplying (2.64) by h^{2k-d} gives

$$\begin{aligned} \sum_{j=1}^n (D^\gamma \ell_j^1(\mathbf{x})) \lambda_i^x \lambda_j^y (h^{2k-d} \Delta^{2k-d}) + h^{2k-d} \sum_{l=1}^q (D^\gamma v_l^1(\mathbf{x})) \lambda_i^x p_l(\mathbf{x}) \\ = \lambda_i^y (h^{2k-d} \tilde{r}_{\gamma,k} \Delta^{2k-d-2|\gamma|}) \end{aligned} \quad (2.65)$$

which can be written as

$$\begin{aligned} h^{|\gamma|} \left\{ \sum_{j=1}^n (h^{-|\gamma|} D^\gamma \ell_j^1(\mathbf{x})) \lambda_i^x \lambda_j^y \phi_{d,k}(\hat{\Delta}) + h^{2k-d-|\gamma|} \sum_{l=1}^q (D^\gamma v_l^1(\mathbf{x})) \lambda_i^x p_l(\mathbf{x}) \right\} \\ = h^{|\gamma|} \lambda_i^y (h^{2k-d-|\gamma|} \tilde{r}_{\gamma,k} \Delta^{2k-d-2|\gamma|}), \end{aligned} \quad (2.66)$$

where $\hat{\Delta} = h\Delta$. Thus

$$\sum_{j=1}^n (h^{-|\gamma|} D^\gamma \ell_j^1(\mathbf{x})) \lambda_i^x \lambda_j^y \phi_{d,k}(\hat{\Delta}) + h^{2k-d-|\gamma|} \sum_{l=1}^q (D^\gamma v_l^1(\mathbf{x})) \lambda_i^x p_l(\mathbf{x}) = \lambda_i^y \left(D^\gamma \phi_{d,k}(\hat{\Delta}) \right). \quad (2.67)$$

If we let

$$\tilde{v}_l^{h\gamma}(\mathbf{x}) = h^{2k-d-|\gamma|} D^\gamma v_l^1(\mathbf{x}), \quad l = 1, \dots, q,$$

then the vector

$$\begin{bmatrix} h^{-|\gamma|} D^\gamma \ell^1(\mathbf{x}) \\ \tilde{v}^{h\gamma}(\mathbf{x}) \end{bmatrix}$$

solves the linear system (2.63) for any $h > 0$. Since the solution of (2.63) is unique, we conclude

$$D^\gamma \ell^h(h\mathbf{x}) = h^{-|\gamma|} D^\gamma \ell^1(\mathbf{x}).$$

When d is even, the i -th row of (2.63), for $h = 1$, is given as

$$\sum_{j=1}^n (D^\gamma \ell_j^1(\mathbf{x})) \lambda_i^x \lambda_j^y (\Delta^{2k-d} \log \Delta) + \sum_{l=1}^q (D^\gamma v_l^1(\mathbf{x})) \lambda_i^x p_l(\mathbf{x}) = \lambda_i^y \left((q_{\gamma,k} + r_{\gamma,k} \log \Delta) \Delta^{2k-d-2|\gamma|} \right). \quad (2.68)$$

Furthermore, if we take $\tilde{t}_i(\mathbf{x})$ as the solution of

$$\sum_{j=1}^n (D^\gamma \ell_j^1(\mathbf{x})) \lambda_i^x \lambda_j^y \Delta^{2k-d} + \tilde{t}_i(\mathbf{x}) = \lambda_i^y D^\gamma \Delta^{2k-d}, \quad (2.69)$$

then combining $h^{2k-d} \times (2.68)$ with $h^{2k-d} \log(h) \times (2.69)$ we have

$$\begin{aligned} & \sum_{j=1}^n (D^\gamma \ell_j^1(\mathbf{x})) \left(h^{2k-d} \lambda_i^x \lambda_j^y (\Delta^{2k-d} \log \Delta) + h^{2k-d} \log(h) \lambda_i^x \lambda_j^y \Delta^{2k-d} \right) \\ & \quad + h^{2k-d} \sum_{l=1}^q (D^\gamma v_l^1(\mathbf{x})) \lambda_i^x p_l(\mathbf{x}) + h^{2k-d} \log(h) \tilde{t}_i(\mathbf{x}) \\ & = h^{2k-d} \lambda_i^y \left((q_{\gamma,k} + r_{\gamma,k} \log \Delta) \Delta^{2k-d-2|\gamma|} \right) + h^{2k-d} \log(h) \lambda_i^y \left(r_{\gamma,k} \Delta^{2k-d-2|\gamma|} \right). \quad (2.70) \end{aligned}$$

Further simplification gives

$$\begin{aligned}
& \sum_{j=1}^n (D^\gamma \ell_j^1(\mathbf{x})) (\lambda_i^x \lambda_j^y \{h^{2k-d} \Delta^{2k-d} (\log(h) + \log \Delta)\}) \\
& \quad + h^{2k-d} \sum_{l=1}^q (D^\gamma v_l^1(\mathbf{x})) \lambda_i^x p_l(\mathbf{x}) + h^{2k-d} \log(h) \tilde{t}_i(\mathbf{x}) \\
& = \lambda_i^y (h^{2k-d} (q_{\gamma,k} + r_{\gamma,k} \log \Delta) \Delta^{2k-d-2|\gamma|} + h^{2k-d} \log(h) r_{\gamma,k} \Delta^{2k-d-2|\gamma|}) \\
& = h^{|\gamma|} \lambda_i^y (h^{2k-d-|\gamma|} \{(q_{\gamma,k} + r_{\gamma,k} \log \Delta) \Delta^{2k-d-2|\gamma|} + \log(h) r_{\gamma,k} \Delta^{2k-d-2|\gamma|}\}) \quad (2.71)
\end{aligned}$$

which we can write as

$$\begin{aligned}
& \sum_{j=1}^n (h^{-|\gamma|} D^\gamma \ell_j^1(\mathbf{x})) \lambda_i^x \lambda_j^y \phi_{d,k}(\hat{\Delta}) + h^{2k-d-|\gamma|} \sum_{l=1}^q (D^\gamma v_l^1(\mathbf{x})) \lambda_i^x p_l(\mathbf{x}) \\
& \quad + h^{2k-d-|\gamma|} \log(h) \tilde{t}_i(\mathbf{x}) = \lambda_i^y (D^\gamma \phi_{d,k}(\hat{\Delta})). \quad (2.72)
\end{aligned}$$

If we set

$$\begin{aligned}
\tilde{v}_1^{h\gamma} &= h^{2k-d-|\gamma|} D^\gamma v_1^1(\mathbf{x}) + h^{2k-d-|\gamma|} \log(h) \tilde{t}_i(\mathbf{x}) \quad \text{and} \\
\tilde{v}_l^{h\gamma} &= h^{2k-d-|\gamma|} D^\gamma v_l^1(\mathbf{x}), \quad l = 2, \dots, q,
\end{aligned}$$

then the vector

$$\begin{bmatrix} h^{-|\gamma|} D^\gamma \ell^1(\mathbf{x}) \\ \tilde{v}^{h\gamma}(\mathbf{x}) \end{bmatrix}$$

solves the linear system (2.63) for any $h > 0$. Once again, since the solution of (2.63) is unique, we conclude

$$D^\gamma \ell^h(h\mathbf{x}) = h^{-|\gamma|} D^\gamma \ell^1(\mathbf{x}).$$

□

A note on local approximation order

We now make a generalization of the approximation order of the local polyharmonic spline interpolant and its derivatives with respect to C^k . This generalises the result in Subsection 2.3.1.

For $u \in C^k$, the k -th order Taylor polynomial of u around $h\mathbf{x}$ is given as

$$T_k(\mathbf{y}) = \sum_{|\alpha| < k} \frac{1}{\alpha!} D^\alpha u(h\mathbf{x}) (\mathbf{y} - h\mathbf{x})^\alpha \quad (2.73)$$

which yields

$$u(h\mathbf{x}) = T_k(\mathbf{y}) - \sum_{0 < |\alpha| < k} \frac{1}{\alpha!} D^\alpha u(h\mathbf{x})(\mathbf{y} - h\mathbf{x})^\alpha. \quad (2.74)$$

By using the representation (2.62) for s^h and the polynomial reproduction property of the Lagrange basis functions, we obtain

$$D^\gamma(u(h\mathbf{x}) - s^h(h\mathbf{x})) = \sum_{i=1}^n D^\gamma \ell_i^h(h\mathbf{x}) \overbrace{[\lambda_i^x T_k(h\mathbf{x}) - \lambda_i^x u(h\mathbf{x})]}{=\mathcal{O}(h^k)}. \quad (2.75)$$

By Lemma 2.22

$$D^\gamma \ell^h(h\mathbf{x}) = h^{-|\gamma|} D^\gamma \ell^1(\mathbf{x}), \quad \text{for } 0 \leq |\gamma| \leq k,$$

and we conclude that

$$|D^\gamma u(h\mathbf{x}) - D^\gamma s^h(h\mathbf{x})| = \mathcal{O}(h^{k-|\gamma|}), \quad h \rightarrow 0. \quad (2.76)$$

Stable evaluation of derivatives

In Subsection 2.3.1, we presented a stable algorithm for the evaluation of polyharmonic splines. Since the ADER-WENO scheme in Chapter 4 involves the derivatives of the polyharmonic splines, we propose a stable way of evaluating the derivatives using the same arguments found in Subsection 2.3.1. Starting with the Lagrange representation of $D^\gamma s^h$ in (2.62), we obtain

$$\begin{aligned} D^\gamma s^h(h\mathbf{x}) &= \langle D^\gamma \ell^h(h\mathbf{x}), u|_{h\Xi} \rangle = \langle h^{-|\gamma|} D^\gamma \ell^1(\mathbf{x}), u_{h\Xi} \rangle \\ &= h^{-|\gamma|} \langle D^\gamma \nu^1(\mathbf{x}), u_h \rangle = h^{-|\gamma|} \langle \mathbf{A}_1^{-1} D^\gamma \beta_1(\mathbf{x}), u_h \rangle \\ &= h^{-|\gamma|} \langle D^\gamma \beta_1(\mathbf{x}), \mathbf{A}_1^{-1} u_h \rangle. \end{aligned} \quad (2.77)$$

The last expression gives a stable evaluation of the derivative of a polyharmonic spline interpolant, which has proven essential in practical computations, as the interpolation matrices can be ill-conditioned in certain cases.

We remark that we do not present any stable evaluation of the RBF interpolant with $\phi_1(r) = r$ which we will use later because we did not experience any serious ill-conditioning of its interpolation matrix.

The computations in this thesis revealed that the smoother the RBF, the worse the conditioning of the interpolation matrix and the greater the need for preconditioning. This agrees with the *uncertainty principle* of Schaback [100] which states that there is no commonly used radial basis function which combines good approximation behaviour with a small condition number of the interpolation matrix.

Chapter 3

The RK-WENO Method

The RK-WENO scheme is a high order finite volume method designed for problems with piecewise smooth solutions containing discontinuities. The RK-WENO methods in [38, 59, 79, 81, 97, 107, 126, 139, 149] and many other related papers are based on polynomial reconstruction methods. Despite the fact that polynomial recovery has the important advantage of being simple to implement and easy to compute, there are some difficulties that arise with this kind of recovery. It is well known from numerical experiments that polynomial reconstruction may lead to numerical instabilities [1]. Although several alternative reconstructions have been proposed [1, 6, 115, 124], both lack of numerical stability and high computational complexity are still critical points for the WENO reconstruction technique, especially for unstructured meshes. Furthermore, when higher degree polynomials are used for reconstruction, the number of coefficients increases significantly. Thus on unstructured grids, finding an admissible stencil of the required size for interpolation may become a difficult task. Moreover, the size of the stencils used for polynomial reconstruction is not always flexible but usually determined by the dimension of the space it belongs to. In fact, to the best of our knowledge, there is no known simple geometrical property that determines the admissibility of a stencil for polynomials of degree greater than one.

In this chapter, we propose an RK-WENO finite volume method on conforming unstructured triangulations where the reconstruction is implemented using radial basis functions (particularly polyharmonic splines), rather than polynomials. The method is based on the theory of optimal recovery [86], whereby polyharmonic splines are identified as optimal recovery functions in Beppo-Levi spaces and yield stable and flexible reconstructions. The necessary oscillation indicators required in the WENO method can be defined naturally using the native Beppo-Levi norms. The RBF reconstruction method admits flexible stencil sizes which makes it easier to obtain admissible stencils. The RBF method is suitable for reconstruction on unstructured grids and is generally not sensitive to the geometry of the grid. Although, the RBF reconstruction also has

the advantage of being suitable in any space dimension, for the sake of simplicity, we will only implement the RBF reconstruction in two space dimensions in this thesis.

3.1 An Introduction to the Finite Volume Method

The finite volume method is based on the integral form of a conservation law (1.3) - (1.4) instead of the differential equation. The method takes full advantage of an arbitrary mesh where there are several options available for the definition of control volumes around which the conservation law can be solved. The method has considerable flexibility, since it allows the modification of the shape and location of the control volumes, as well as variation of the rules for flux evaluation [74, 84].

In the finite volume method, the spatial domain, $\Omega \subset \mathbb{R}^d$, is first divided into a collection of control volumes that completely cover the domain. We shall refer to the control volumes as cells, elements or triangles at various points in this work. If we let \mathcal{T} denote a tessellation of a domain Ω with control volumes $T \in \mathcal{T}$ such that $\cup_{T \in \mathcal{T}} \bar{T} = \bar{\Omega}$, then in each control volume, the integral form of the conservation law is defined as

$$\frac{d}{dt} \int_T u d\mathbf{x} + \int_{\partial T} F(u) \cdot \mathbf{n} ds = 0. \quad (3.1)$$

The integral conservation law is readily obtained upon the spatial integration of (1.3) on T and application of the divergence theorem.

Central to the finite volume method is the definition of the cell average for each control volume $T \in \mathcal{T}$

$$\bar{u}_T = \frac{1}{|T|} \int_T u d\mathbf{x}. \quad (3.2)$$

The flux integral in (3.1) can be approximated by

$$\int_{\partial T} F(u) \cdot \mathbf{n} ds \approx \sum_{\Gamma_{TR} \in \partial T} \mathcal{F}(\bar{u}_T, \bar{u}_R; \mathbf{n}), \quad (3.3)$$

where \mathcal{F} is the numerical flux and R is the neighbouring control volume sharing the edge Γ_{TR} with T with outer normal \mathbf{n} .

The semi-discrete finite volume method is then obtained by dividing (3.1) by $|T|$, yielding the numerical method

$$\frac{d}{dt} \bar{u}(t) = -\frac{1}{|T|} \sum_{\Gamma_{TR} \in \partial T} \mathcal{F}_{TR}(\bar{u}_T^n, \bar{u}_R^n; \mathbf{n}), \quad \bar{u}_T^0 = \frac{1}{|T|} \int_T u_0(\mathbf{x}) d\mathbf{x}, \quad \text{for all } T \in \mathcal{T}, \quad (3.4)$$

where for all \mathcal{F} , we assume that for any $L > 0$ and for all $u, v, u', v' \in B_L(0)$ we have

$$|\mathcal{F}(u, v; \mathbf{n}) - \mathcal{F}(u', v'; \mathbf{n})| \leq c(L)h(|u - u'| + |v - v'|), \quad h = \sup_{T \in \mathcal{T}} \text{diam}(T), \quad (3.5)$$

$$\mathcal{F}(u, v; \mathbf{n}) = -\mathcal{F}(v, u; \mathbf{n}), \quad (3.6)$$

$$\mathcal{F}(u, u; \mathbf{n}) = \int_{\Gamma_{TR}} F(u) \cdot \mathbf{n} \, ds. \quad (3.7)$$

The condition (3.5) is a local Lipschitz condition, (3.6) is the conservation property and (3.7) consistency. The set of ordinary differential equations (3.4) can be advanced in time using a number of implicit or explicit multi-step or Runge-Kutta methods. For instance, a simple time-stepping method is the forward Euler method which produces the fully discrete method

$$\bar{u}_T^{n+1} = \bar{u}_T^n - \frac{\Delta t}{|T|} \sum_{\Gamma_{TR} \in \partial T} \mathcal{F}(\bar{u}_T^n, \bar{u}_R^n; \mathbf{n}), \quad \text{for all } T \in \mathcal{T}. \quad (3.8)$$

High order finite volume methods are constructed by using piecewise recovery functions that are not constant on each cell instead of the cell averages (which are piecewise constant) in computing the numerical fluxes. These recovery functions are obtained from the cell averages using a suitable reconstruction technique.

3.1.1 Numerical fluxes

Numerical fluxes play a key role in the design of finite volume methods. Unfortunately, the numerical flux conditions (3.5) - (3.7) do not necessarily guarantee convergence to entropy satisfying weak solutions and additional numerical flux restrictions may be necessary [9, 74]. The best-known examples of numerical fluxes \mathcal{F} that satisfy properties (3.5), (3.6), (3.7) and guarantee convergence are

(i) The Engquist-Osher flux:

$$\mathcal{F}^{EO}(u_l, u_r; \mathbf{n}) = \frac{F(u_l) + F(u_r)}{2} \cdot \mathbf{n} + \frac{1}{2} \int_{u_r}^{u_l} |F'(u) \cdot \mathbf{n}| \, du;$$

(ii) The Lax-Friedrichs flux:

$$\mathcal{F}^{LF}(u_l, u_r; \mathbf{n}) = \frac{1}{2} [F(u_l) + F(u_r) - \sigma(u_r - u_l)],$$

$$\sigma = \max_{\inf u_0(\mathbf{x}) \leq u \leq \sup u_0(\mathbf{x})} |F'(u) \cdot \mathbf{n}|;$$

(iii) The local Lax-Friedrichs flux:

$$\mathcal{F}^{LLF}(u_l, u_r; \mathbf{n}) = \frac{1}{2} [F(u_l) + F(u_r) - \sigma(u_r - u_l)],$$

$$\sigma = \max_{\min(u_l, u_r) \leq u \leq \max(u_l, u_r)} |F'(u) \cdot \mathbf{n}|;$$

(iv) The Roe flux with entropy:

$$\mathcal{F}^{ROE}(u_l, u_r; \mathbf{n}) = \frac{F(u_l) + F(u_r)}{2} \cdot \mathbf{n} + |A(u_l, u_r)| \frac{u_l - u_r}{2};$$

where

$$A(u_l, u_r) = \begin{cases} F'(u) \cdot \mathbf{n}, & u_l = u_r = u; \\ \frac{F(u_l) - F(u_r)}{u_l - u_r} \cdot \mathbf{n}, & u_l \neq u_r. \end{cases}$$

Numerical experiments have shown that as the degree of the approximation increases the choice of the numerical flux does not have any significant effect on the quality of approximation [10].

3.2 Finite Volume Formulation on Triangular Meshes

In the rest of this thesis, we will be solving numerically the two dimensional conservation law

$$\frac{\partial u}{\partial t} + \nabla \cdot F(u) = 0, \quad (3.9)$$

by using a finite volume discretisation on a computational domain $\Omega \in \mathbb{R}^2$ with polygonal boundary, and for a compact time interval, subject to appropriate initial and boundary conditions.

We begin by making the above discussion more precise.

3.2.1 Conforming triangulations

Definition 3.1 ([84]) *A triangulation \mathcal{T} of Ω is the finitely many subsets $T_i \subset \mathcal{T}$, $i = 1, \dots, \#\mathcal{T}$, such that the following conditions are satisfied:*

- $\Omega = \bigcup_{i \in \{1, \dots, \#\mathcal{T}\}} T_i$.
- Every $T_i \in \mathcal{T}$ is closed and the interior of T_i is not empty.
- For any two $T_i, T_j \in \mathcal{T}$ with $i \neq j$, $T_i \cap T_j$ is of empty interior.
- The boundary of every $T_i \in \mathcal{T}$ is Lipschitz continuous.

Furthermore, a triangulation is called **conforming**, if

- Every one-dimensional edge of any $T_i \in \mathcal{T}$ is either a subset of the boundary $\partial\Omega$ or the edge of another T_j , $j \neq i$.

Conformity ensures that no hanging nodes occur in the triangulation. Definition 3.1 does not necessarily require that the T_i 's be triangles. Fortunately, finite volume discretisations are quite flexible and can be implemented on several types of triangulations and even on hybrid meshes. In this work, we shall implement our finite volume methods only on conforming triangular grids.

Definition 3.2 Any set of control volumes

$$\mathcal{S}(T_i) := \{T_{i_1}, T_{i_2}, \dots, T_{i_N}\}$$

of size N is called a **stencil** for T_i , if $i_1 \equiv i$ and $T_{i_j} \neq T_{i_k}$ for $j \neq k$ holds.

3.2.2 Semi-discrete formulation on triangular meshes

For any triangle $T \in \mathcal{T}$, the semi-discrete scheme, based on the integral form of (3.9), has the form

$$\frac{d}{dt} \bar{u}_T + \frac{1}{|T|} \int_{\partial T} F(u) \cdot \mathbf{n} \, ds = 0, \quad \text{for } T \in \mathcal{T}, \quad (3.10)$$

where

$$\bar{u}_T \equiv \bar{u}_T(t) = \frac{1}{|T|} \int_T u(t, \mathbf{x}) \, d\mathbf{x}, \quad \text{for } T \in \mathcal{T}, t \in I, \quad (3.11)$$

denotes the cell average of u over triangle $T \in \mathcal{T}$ at time $t \in I$, \mathbf{n} is the outward unit normal vector of the triangle's boundary ∂T and $|T|$ is the area of triangle T .

The boundary ∂T of triangle $T \in \mathcal{T}$ is given by the union of three edges, say $\Gamma_1, \Gamma_2, \Gamma_3$, in the triangulation \mathcal{T} , i.e.

$$\partial T = \bigcup_{j=1}^3 \Gamma_j,$$

so that the line integral in (3.10) can be represented as

$$\int_{\partial T} F \cdot \mathbf{n} \, ds = \sum_{j=1}^3 \int_{\Gamma_j} F(u(t, s)) \cdot \mathbf{n}_j \, ds, \quad (3.12)$$

where \mathbf{n}_j is the outward unit normal vector for edge Γ_j . We discretise the integral on the right hand side of (3.12) by using a q -point Gaussian integration formula, for some specific $q \in \mathbb{N}$ which determines the order of the resulting quadrature rule.

Let G_1, \dots, G_q and w_1, \dots, w_q denote the Gaussian points and weights for the trian-

gle's edge Γ_j . Then, the Gaussian quadrature formula

$$\int_{\Gamma_j} F(u(t, s)) \cdot \mathbf{n}_j ds \approx |\Gamma_j| \sum_{\ell=1}^q w_\ell F(u(t, G_\ell)) \cdot \mathbf{n}_j, \quad j = 1, 2, 3,$$

yields an order $2q - 1$ approximation to the line integral (3.12), and so (3.10) is replaced by

$$\frac{d}{dt} \bar{u}_T(t) + \frac{1}{|T|} \sum_{j=1}^3 |\Gamma_j| \sum_{\ell=1}^q w_\ell F(u(t, G_\ell)) \cdot \mathbf{n}_j = 0. \quad (3.13)$$

Finally, we replace the terms $F(u(t, G_\ell)) \cdot \mathbf{n}_j$, $1 \leq \ell \leq q$, by a numerical flux function to approximate the flux across the boundary of neighbouring triangles to $T \in \mathcal{T}$. In this work, we use the Lax-Friedrichs flux, given by

$$F(u(t, G_\ell)) \cdot \mathbf{n} \approx \mathcal{F}(u_{\text{in}}(t, G_\ell), u_{\text{out}}(t, G_\ell); \mathbf{n}) = \frac{1}{2} [(F(u_{\text{in}}(t, G_\ell)) + F(u_{\text{out}}(t, G_\ell))) \cdot \mathbf{n} - \sigma(u_{\text{in}}(t, G_\ell) - u_{\text{out}}(t, G_\ell))], \quad (3.14)$$

where σ is an upper bound for the eigenvalues of the flux function's Jacobian matrix in the normal direction \mathbf{n} . Moreover, for time $t \in I$, $u_{\text{in}}(t, G_\ell)$ in (3.14) is the function value of the solution's representation over triangle T , and $u_{\text{out}}(t, G_\ell)$ is the function value of the corresponding representation over the neighbouring triangle that shares the edge Γ_j with T .

Thus (3.13) is replaced by

$$\frac{d}{dt} \bar{u}_T(t) = L_T(\bar{u}_T(t)), \quad \text{for } T \in \mathcal{T}, \quad (3.15)$$

where

$$L_T(\bar{u}_T(t)) = -\frac{1}{|T|} \sum_{j=1}^3 |\Gamma_j| \sum_{\ell=1}^q w_\ell \mathcal{F}(u_{\text{in}}(t, G_\ell), u_{\text{out}}(t, G_\ell)) \cdot \mathbf{n}_j, \quad (3.16)$$

for some specific univariate function L_T .

The remaining steps are the time discretisation and reconstruction from cell averages. The reconstruction enables us obtain u_{in} and u_{out} that are used in computing the numerical flux (3.14). We will first of all describe the time stepping algorithm before proceeding with the reconstruction technique which is the main theme of this chapter.

3.3 Time Discretisation

Equation (3.15) is a finite set of ordinary differential equations (ODEs) which has to be discretised by a suitable ODE solver. Due to the nature of hyperbolic conservation laws,

it is important to select the appropriate time discretisation to use. Our choice of time discretisations for (3.15) is the class of high-order Strong Stability Preserving Runge-Kutta (SSPRK) methods which first appeared in the work of Shu & Osher [108, 110] and were referred to as TVD (Total Variation Diminishing) Runge-Kutta methods. They have been further investigated and tested in [43, 44, 98, 119, 140]. The use of this class of methods is important because numerical experiments in [44, 45] show that oscillations may occur when high order time discretisation methods that lack the strong stability preserving property are used in solving (3.15).

The idea of the SSP methods is to assume that the first order forward Euler time discretisation of the set of ODEs (3.15) is strongly stable under a certain norm when the time step Δt is suitably restricted, and then to try to find a higher order time discretisation (Runge-Kutta or multistep) that maintains strong stability for the same norm, perhaps under a different time size restriction [44]. In [108, 110], the relevant norm was the total variation norm. The infinity norm can also be another natural possibility.

3.3.1 The CFL condition

It is well known that explicit time stepping methods have to satisfy a restriction on the time step Δt due to the Courant-Friedrichs-Lewy (CFL) condition. This condition, in general, requires that during one step, information from one control volume must not interact with information coming from other control volumes.

The CFL condition we use in all our numerical experiments is the same one used in [73] which is similar to the CFL condition proposed in [84]. If we let r_T to be radius of the inscribed circle of triangular cell $T \in \mathcal{T}$ and if

$$\zeta_T^{(\max)} = \max_{1 \leq j \leq 3q} |f'_{1,j}(u)n_{1,j} + f'_{2,j}(u)n_{2,j}|$$

denotes the maximum normal characteristic speed at the $3q$ Gaussian integration points of the three cell edges, then the time step Δt on an unstructured triangular mesh \mathcal{T} is restricted by the CFL condition

$$\Delta t \leq c \cdot \min_{T \in \mathcal{T}} \frac{r_T}{\zeta_T^{(\max)}}$$

with a CFL constant c .

3.3.2 Optimal SSP Runge-Kutta methods

In [44, 110], optimal (in the sense of CFL coefficient and the cost incurred by additional operator computations) SSP Runge-Kutta methods were developed and analysed. In

this thesis, we advance the cell averages from t^n to t^{n+1} by solving (3.15) using the SSPRK(2, 2) and the SSPRK(3, 3) methods, where the notation SSPRK(s, p) is used to denote an s -stage and p -th order method. If we let \bar{u}_T^n and \bar{u}_T^{n+1} denote the finite volume approximations at $t = t^n$ and $t = t^{n+1}$ respectively for any $T \in \mathcal{T}$, then the optimal SSPRK(2, 2), according to Shu & Gottlieb [44], is given as

$$\begin{aligned}\bar{u}_T^{(1)} &= \bar{u}_T^n + \Delta t L_T(\bar{u}_T^n), \\ \bar{u}_T^{n+1} &= \frac{1}{2}\bar{u}_T^n + \frac{1}{2}\bar{u}_T^{(2)} + \frac{1}{2}\Delta t L_T(\bar{u}_T^{(2)}),\end{aligned}\tag{3.17}$$

while the optimal SSPRK(3, 3) is given as

$$\begin{aligned}\bar{u}_T^{(1)} &= \bar{u}_T^n + \Delta t L_T(\bar{u}_T^n), \\ \bar{u}_T^{(2)} &= \frac{3}{4}\bar{u}_T^n + \frac{1}{4}\bar{u}_T^{(1)} + \frac{1}{4}\Delta t L_T(\bar{u}_T^{(1)}), \\ \bar{u}_T^{n+1} &= \frac{1}{3}\bar{u}_T^n + \frac{2}{3}\bar{u}_T^{(2)} + \frac{2}{3}\Delta t L_T(\bar{u}_T^{(2)}).\end{aligned}\tag{3.18}$$

We note that the function L_T is recomputed for each time step and the SSPRK method only advances the solution from one time step to another; the superscripts in parenthesis denote the stage number of the scheme.

The SSPRK(3, 3) is probably the most commonly used SSP Runge-Kutta method. Despite the fact that it is only third order accurate, it is used even when the spatial accuracy is much larger than three. Higher order in time is then achieved by using a smaller time step that is an appropriate power of the mesh size, e.g. [59]. This ensures that the order of the time discretisation, if lower than the order of the spatial discretisation, does not dominate the overall convergence rate of the method. For example, when SSPRK(3, 3) is used with a fourth order spatial reconstruction, one can use $\Delta t = (\Delta x)^{4/3}$ to get a method that is of fourth order accuracy. The popularity of SSPRK(3, 3) is a result of its simplicity, its classical stability analysis and the fact that finding a suitable fourth or fifth order SSP Runge-Kutta method has proved to be a very difficult task [43, 44, 45]. Moreover, fourth and fifth order methods require extra operator computations and have more severe CFL restrictions.

In Chapter 4, we will implement an alternative time stepping strategy with the ADER-WENO method. It is based on Taylor series in time in combination with a Cauchy-Kowalewski (Lax-Wendroff) procedure and the solution of generalized Riemann problems.

3.4 Reconstruction of Polyharmonic Splines from Cell Averages

We will first of all look at the general case of reconstruction with radial basis functions, before looking at the specific instance of reconstruction with polyharmonic splines considered in this work. Other choices of RBFs could be used in the algorithms described in this thesis, so we prefer to keep the discussion general whenever possible.

3.4.1 Reconstruction by radial basis functions

Given a conforming triangulation $\mathcal{T} = \{T\}_{T \in \mathcal{T}}$ and a triangle $T \in \mathcal{T}$, consider a stencil

$$\mathcal{S} = \{T_1, T_2, \dots, T_n\} \subset \mathcal{T}$$

of size $\#\mathcal{S} = n$, containing T , i.e. $T \in \mathcal{S}$. Suppose the triangles in stencil \mathcal{S} are associated with the linearly independent functionals $\{\lambda_T\}_{T \in \mathcal{S}}$,

$$\lambda_T(u) = \frac{1}{|T|} \int_T u(\mathbf{x}) d\mathbf{x}, \quad \text{for } T \in \mathcal{T} \quad \text{and} \quad u(\mathbf{x}) \equiv u(t, \mathbf{x}),$$

i.e. the cell average operator for triangle T .

Given the cell averages $\{\lambda_T(u)\}_{T \in \mathcal{S}}$ for any stencil $\mathcal{S} \subset \mathcal{T}$, we consider solving the reconstruction problem

$$\lambda_T(u) = \lambda_T(s), \quad \text{for all } T \in \mathcal{S}, \quad (3.19)$$

where

$$s(\mathbf{x}) = \sum_{T \in \mathcal{S}} c_T \lambda_T^y \phi(\|\mathbf{x} - \mathbf{y}\|) + p(\mathbf{x}), \quad p \in \mathcal{P}_m^d, \quad (3.20)$$

is the form of the reconstruction s with $\phi : [0, \infty) \rightarrow \mathbb{R}$ is a fixed radial basis function and $\|\cdot\|$ is the Euclidean norm on \mathbb{R}^d and where \mathcal{P}_m^d is defined as the vector space of all d -variate polynomials of degree at most $m - 1$. Recall that the dimension of \mathcal{P}_m^d is $q = \dim(\mathcal{P}_m^d) = \binom{m-1+d}{d}$. Moreover, λ_T^y in (3.20) denotes the action of the linear functional λ_T with respect to the variable \mathbf{y} , i.e.

$$\lambda_T^y \phi(\|\mathbf{x} - \mathbf{y}\|) = \frac{1}{|T|} \int_T \phi(\|\mathbf{x} - \mathbf{y}\|) d\mathbf{y}.$$

Possible choices for ϕ along with their order m are shown in Table 2.1.

Furthermore, the reconstruction s in (3.20) contains $n + q$ parameters, (n for its major part and q for its polynomial part) but at only $n = \#\mathcal{S}$ interpolation conditions

in (3.19). To eliminate the additional q degrees of freedom, we need to solve (3.19) under linear constraints

$$\sum_{T \in \mathcal{S}} c_T \lambda_T(p) = 0, \quad \text{for all } p \in \mathcal{P}_m^d, \quad (3.21)$$

where λ_T is the cell average operator of triangle T . This leads us to the $(n+q) \times (n+q)$ linear system

$$\begin{bmatrix} A & P \\ P^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} u|_{\mathcal{S}} \\ 0 \end{bmatrix}, \quad (3.22)$$

where

$$A = (\lambda_T^x \lambda_R^y \phi(\|\mathbf{x} - \mathbf{y}\|))_{T,R \in \mathcal{S}} \in \mathbb{R}^{n \times n} \text{ and } P = (\lambda_T(\mathbf{x}^\alpha))_{T \in \mathcal{S}, 0 \leq |\alpha| < m} \in \mathbb{R}^{n \times q},$$

and $u|_{\mathcal{S}} = (\lambda_T(u))_{T \in \mathcal{S}} \in \mathbb{R}^n$.

The linear system (3.22) has for any radial basis function ϕ in Table 2.1 a unique solution for the unknown coefficients $\mathbf{c} \in \mathbb{R}^n$ (for the major part of s) and $\mathbf{d} \in \mathbb{R}^q$ (for the polynomial part of s), provided that the set $\{\lambda_T\}_{T \in \mathcal{S}}$ of cell average operators is \mathcal{P}_m^d -*unisolvent*, i.e., for $p \in \mathcal{P}_m^d$ we have

$$\lambda_T(p) = 0 \quad \text{for all } T \in \mathcal{T} \quad \implies \quad p \equiv 0,$$

in which case any polynomial from \mathcal{P}_m^d can uniquely be reconstructed from its values $\{\lambda_T(p)\}_{T \in \mathcal{S}}$. This standard result dates back to the seminal work of Micchelli [85].

3.4.2 Reconstruction by polyharmonic splines

In the polyharmonic spline reconstruction method, the radial function $\phi_{d,k}$ is given in (2.16) with order $m = k$. An important example which we will use extensively in this thesis is the case $d = k = 2$ which is the thin plate spline with $\phi_{2,2}(r) = r^2 \log(r)$. In this case, the reconstruction s in (3.20) has the form

$$s(\mathbf{x}) = \sum_{T \in \mathcal{S}} c_T \lambda_T^y(\|\mathbf{x} - \mathbf{y}\|^2 \log(\|\mathbf{x} - \mathbf{y}\|)) + d_1 + d_2 x_1 + d_3 x_2,$$

where we let x_1 and x_2 denote the two coordinates of $\mathbf{x} = (x_1, x_2)^T \in \mathbb{R}^2$.

As stated in Chapter 2, $\phi_{d,k}$ possesses an optimal reconstruction property in the Beppo-Levi space. This variational property, due to Duchon [34], says that for $\phi \equiv \phi_{d,k}$ the reconstruction $s \in \text{BL}_k(\mathbb{R}^d)$ in (3.20) minimises the energy $|\cdot|_{\text{BL}_k(\mathbb{R}^d)}$ among all elements in $\text{BL}_k(\mathbb{R}^d)$ satisfying (3.19). This implies that for any $u \in \text{BL}_k(\mathbb{R}^d)$, we have

$$|s|_{\text{BL}_k(\mathbb{R}^d)} \leq |u|_{\text{BL}_k(\mathbb{R}^d)}, \quad \text{with } u|_{\mathcal{S}} = s|_{\mathcal{S}}, \quad (3.23)$$

where $s \in \text{BL}_k(\mathbb{R}^d)$ is the reconstruction of u from data $\{\lambda_T(u)\}_{T \in \mathcal{S}}$, i.e. s satisfies (3.19).

3.5 Local Lagrange Reconstruction by Polyharmonic Splines

It was shown in the paper of Iske [63] that the implementation of a polyharmonic spline interpolation scheme requires carefully handling due to issues relating to numerical stability. This is mainly because a direct solution of (3.19), (3.21) may lead to coefficient matrices in (3.22), whose spectral condition number is very large in situations when the barycentres of two distinct triangles in stencil $\mathcal{S} \subset \mathcal{T}$ are very close. This important observation, due to Narcowich & Ward [88], motivated the construction of a preconditioner for the linear system (3.22) in [63] which we described in Subsection 2.3.1. Other techniques for preconditioning the RBF interpolation scheme can be found in the literature, e.g. [36, 144]. A detailed treatment of the stability of radial basis function interpolants can be found in the papers of Schaback [100, 102].

In order to use the results in Subsection 2.3.1 on the stable evaluation of the polyharmonic spline interpolant, we will work with a scaled reconstruction problem based on a local Lagrange-type reconstruction. This is appropriate because when constructing high order finite volume methods, one is concerned with local approximation schemes. For the sake of our discussion in this section and in Subsection 4.1.3, we will denote our polyharmonic spline interpolant as s^h instead of s , where we take the parameter h to be the local mesh size of a triangulation \mathcal{T}^h . We also use the superscript h on the triangles and stencils in \mathcal{T}^h . The description of the interpolation problem we present in this section is the same as in Subsection 2.3.1 but using different notations .

Suppose we have a triangle $T^h \in \mathcal{T}^h$ with stencil $\mathcal{S}^h = \{T_1^h, T_2^h, \dots, T_n^h\} \subset \mathcal{T}^h$, i.e. $T^h \in \mathcal{S}^h$. Then for some fixed point $\mathbf{x}_0 \in \mathbb{R}^2$ and any $h > 0$, $\mathbf{x}_0 + h\mathbf{x} \in T^h$ and the *scaled* reconstruction problem is

$$\lambda_{T^h} u(\mathbf{x}_0 + h\mathbf{x}) = \lambda_{T^h} s^h(\mathbf{x}_0 + h\mathbf{x}), \quad T^h \in \mathcal{S}^h, \quad (3.24)$$

where the set of cell averages operators $\{\lambda_{T^h}\}_{T^h \in \mathcal{S}^h}$ is \mathcal{P}_k^2 -*unisolvent* and of moderate size. In addition, s^h denotes the unique polyharmonic spline interpolant of the form

$$s^h(h\mathbf{x}) = \sum_{T^h \in \mathcal{S}^h} c_{T^h}^h \lambda_{T^h}^y \phi_{2,k}(\|h\mathbf{x} - h\mathbf{y}\|) + p(h\mathbf{x}), \quad p \in \mathcal{P}_k^2, \quad (3.25)$$

satisfying (3.24) and where we assume, without loss of generality, that $\mathbf{x}_0 = 0$.

The coefficients $\mathbf{c}^h \in \mathbb{R}^n$, $\mathbf{d}^h = (d_\alpha^h)_{|\alpha| < k} \in \mathbb{R}^q$ of the interpolant s^h in (3.25) are

obtained by solving

$$\underbrace{\begin{bmatrix} A_h & P_h \\ P_h^T & \mathbf{0} \end{bmatrix}}_{\mathbf{A}_h} \underbrace{\begin{bmatrix} \mathbf{c}^h \\ \mathbf{d}^h \end{bmatrix}}_{\mathbf{b}^h} = \underbrace{\begin{bmatrix} u|_{\mathcal{S}^h} \\ 0 \end{bmatrix}}_{u_h}, \quad (3.26)$$

where

$$A_h = (\lambda_{T^h}^x \lambda_{R^h}^y \phi_{2,k}(\|h\mathbf{x} - h\mathbf{y}\|))_{T^h, R^h \in \mathcal{S}^h} \in \mathbb{R}^{n \times n} \text{ and } P_h = (\lambda_{T^h}(h\mathbf{x})^\alpha)_{T^h \in \mathcal{S}^h, 0 \leq |\alpha| < k} \in \mathbb{R}^{n \times q},$$

$$\text{and } u|_{\mathcal{S}^h} = (\lambda_{T^h}(u))_{T^h \in \mathcal{S}^h} \in \mathbb{R}^n.$$

The Lagrange-type representation of the s^h is given as

$$s^h(h\mathbf{x}) = \sum_{T^h \in \mathcal{S}^h} \ell_{T^h}^h(h\mathbf{x}) \lambda_{T^h} u(h\mathbf{x}) \quad (3.27)$$

with the Lagrange basis functions $\ell_{T^h}^h$ satisfying

$$\lambda_{R^h}(\ell_{T^h}^h(h\mathbf{x})) = \begin{cases} 1, & \text{for } T^h = R^h; \\ 0, & \text{for } T^h \neq R^h, \end{cases} \quad T^h, R^h \in \mathcal{S}^h. \quad (3.28)$$

Furthermore, $\ell_{T^h}^h$ is defined in such a way that it reproduces polynomials, i.e.

$$\sum_{T^h \in \mathcal{S}^h} \ell_{T^h}^h(h\mathbf{x}) \lambda_{T^h} p(h\mathbf{x}) = p(h\mathbf{x}), \quad p \in \mathcal{P}_k^2.$$

Moreover, since $\phi_{d,k}$ is conditionally positive definite and if \mathcal{S}^h allows polynomial recovery of degree q , the matrix A_h is regular and thus for all $h\mathbf{x} \in \mathbb{R}^2$, the vector $\ell^h(h\mathbf{x}) = (\ell_{T^h}^h(h\mathbf{x}))_{T^h \in \mathcal{S}^h} \in \mathbb{R}^n$ together with $v^h(h\mathbf{x}) = (v_1^h(h\mathbf{x}), \dots, v_q^h(h\mathbf{x}))^T$ are the unique solution of the linear system

$$\underbrace{\begin{bmatrix} A_h & P_h \\ P_h^T & \mathbf{0} \end{bmatrix}}_{\mathbf{A}_h} \underbrace{\begin{bmatrix} \ell^h(h\mathbf{x}) \\ v^h(h\mathbf{x}) \end{bmatrix}}_{\nu^h(h\mathbf{x})} = \underbrace{\begin{bmatrix} R_h(h\mathbf{x}) \\ S_h(h\mathbf{x}) \end{bmatrix}}_{\beta_h(h\mathbf{x})}, \quad (3.29)$$

where

$$\begin{aligned} R_h(h\mathbf{x}) &= (\lambda_{T^h}^y \phi_{2,k}(\|h\mathbf{x} - h\mathbf{y}\|))_{T^h \in \mathcal{S}^h} \in \mathbb{R}^n, \\ S_h(h\mathbf{x}) &= ((h\mathbf{x})^\alpha)_{|\alpha| < k} \in \mathbb{R}^q. \end{aligned}$$

From Lemma 2.13, we know that

$$\ell^h(h\mathbf{x}) = \ell^1(\mathbf{x}), \quad \text{for every } \mathbf{x} \in \mathbb{R}^d. \quad (3.30)$$

Hence, from (3.30) and using the preconditioning that yielded (2.52) in Chapter 2, the stable evaluation of s^h is given as

$$\begin{aligned} s^h(h\mathbf{x}) &= \langle \ell^h(h\mathbf{x}), u|_{\mathcal{S}^h} \rangle = \langle \ell^1(\mathbf{x}), u|_{\mathcal{S}^h} \rangle \\ &= \langle \nu^1(\mathbf{x}), u_h \rangle = \langle \mathbf{A}_1^{-1} \beta_1(\mathbf{x}), u_h \rangle \\ &= \langle \beta_1(\mathbf{x}), \mathbf{A}_1^{-1} u_h \rangle \end{aligned} \quad (3.31)$$

where ν^1 , \mathbf{A}_1 , ℓ^1 , β_1 are obtained when we set $h = 1$ in ν^h , \mathbf{A}_h , ℓ^h , β_h . We note that the use of (3.31) is not just a theoretical exercise but is actually required in the computation of a stable reconstruction, especially when the mesh gets finer. For thin plate spline reconstruction, the form (3.31), while not essential, did improve the quality of our numerical computations. However, when implementing reconstruction with $\phi_{2,3}(r) = r^4 \log(r)$, this preconditioning was essential for obtaining any useful numerical results.

3.6 Stencil Selection

The WENO reconstruction method requires construction on a number of admissible stencils for each triangle $T \in \mathcal{T}$. The selection strategy for admissible stencils for control volumes on an unstructured grid is an important but difficult task, especially as the stencil size increases. There are usually many possible approaches but care must be taken in which one is used. To this end, some important considerations for selecting n suitable stencils $\mathcal{S}_i, i = 1, \dots, n$, around a control volume $T \in \mathcal{S}_i \subset \mathcal{T}$ were proposed in [38, 72, 116]:

- every stencil should be local (relative to its corresponding center T);
- the number of stencils, n , should be small in order to keep the required computational cost small;
- in smooth regions of the solution the stencils should, for the sake of good approximation quality, be well-centered (i.e. isotropic);
- in non-smooth (i.e. discontinuous) regions of the solution, one-sided (i.e. anisotropic) stencils should be preferred in order to avoid interpolation across discontinuities, which may lead to undesired oscillations.

When the stencil size is not too large, the stencils can be constructed on the basis of two neighbourhoods of triangles.

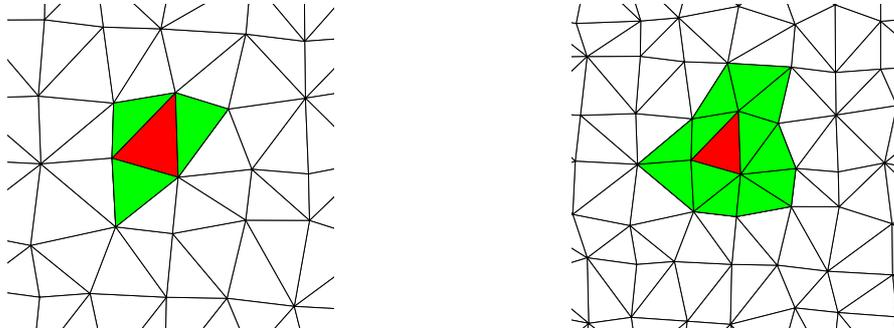
Definition 3.3 ([116]) For triangle $T \in \mathcal{T}$ the set

$$\mathcal{N}_v(T) := \{R \in \mathcal{T} : R \cap T \text{ is an edge of } T \text{ and } T \neq R\}$$

is called the **von Neumann neighbourhood** (there are at most three von Neumann neighbours) of T . The set

$$\mathcal{N}_M(T) := \{R \in \mathcal{T} : R \cap T \text{ is an edge of } T \text{ or a node of } T\}$$

is called a **Moore neighbourhood** of T .



(a) von Neumann neighbourhood

(b) Moore neighbourhood

Figure 3.1: Two neighbourhoods of the triangle T (marked red).

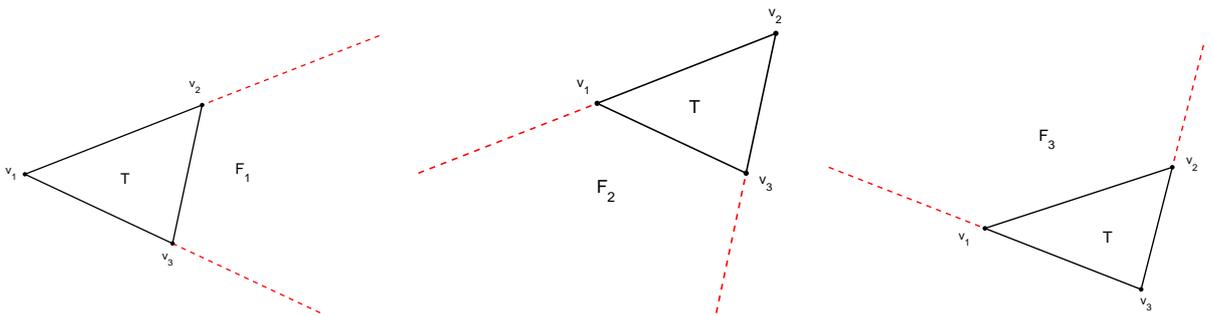
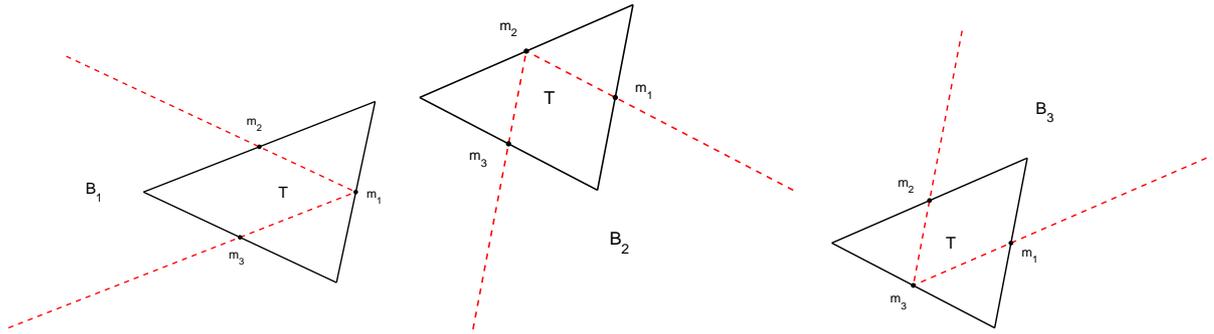


Figure 3.2: The three forward sectors of triangle T .

These two neighbourhoods will be inadequate if larger stencils are required. To this end, in [38], three *centred* stencils for T were constructed by extending its von Neumann neighbourhood. This was achieved by adding some of the von Neumann neighbours of the original von Neumann neighbours of T to the stencil until the desired stencil size was achieved. Additional stencils can be found by using a sectoral search algorithm

Figure 3.3: The three backward sectors of triangle T .

as proposed by Harten & Chakravarthy [51]. This is particularly useful in the non-smooth regions of the solution. The idea in [51] is to include the extended von Neumann neighbours of a triangle $T \in \mathcal{T}$, whose barycentres lie in one of the three *forward sectors* F_j , $j = 1, 2, 3$ (defined below) of T .

Definition 3.4 ([116]) *Let $T \in \mathcal{T}$ be a triangle with vertices $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$. The sets*

$$F_j = \{\mathbf{x} = \mathbf{v}_j + \alpha_1 \mathbf{f}_{j1} + \alpha_2 \mathbf{f}_{j2} : \alpha_1, \alpha_2 \geq 0\}, \quad j = 1, 2, 3$$

defined by the vector pairs ($\mathbf{f}_{11} = \mathbf{v}_2 - \mathbf{v}_1, \mathbf{f}_{12} = \mathbf{v}_3 - \mathbf{v}_1$), ($\mathbf{f}_{21} = \mathbf{v}_3 - \mathbf{v}_2, \mathbf{f}_{22} = \mathbf{v}_1 - \mathbf{v}_2$) and ($\mathbf{f}_{31} = \mathbf{v}_1 - \mathbf{v}_3, \mathbf{f}_{32} = \mathbf{v}_2 - \mathbf{v}_3$) are called the forward sectors of T , see Figure 3.2.

The resulting one-sided stencils are known as forward stencils. In [72], further improvement was made in the construction of one-sided stencils by using additional sectors, called *backward sectors*. These additional sectors cover regions around a triangle that are not covered by the centred stencils or the forward sectors. For any triangle T , its three backward sectors B_j , $j = 1, 2, 3$, are obtained by using the three midpoints $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3$ of the edges of T . Each backward sector has its origin at one midpoint and its boundary edges passing through the other two midpoints as shown in Figure 3.3. The three backward stencils then include the extended von Neumann neighbours of T that lie in the corresponding backward sector. For any triangle $T \in \mathcal{T}$, each of its three backward sectors, B_j , corresponds to an opposite forward sector F_j , $j = 1, 2, 3$. The geometry of the complementary six sectors, F_j and B_j , $j = 1, 2, 3$, allow us to obtain preference directions of the solution around triangle T , which in turn improves the quality of the reconstruction on T .

Figures 3.4, 3.5 and 3.6 show three centred stencils, three forward stencils and three backward stencils of size five for the triangle T (marked red).

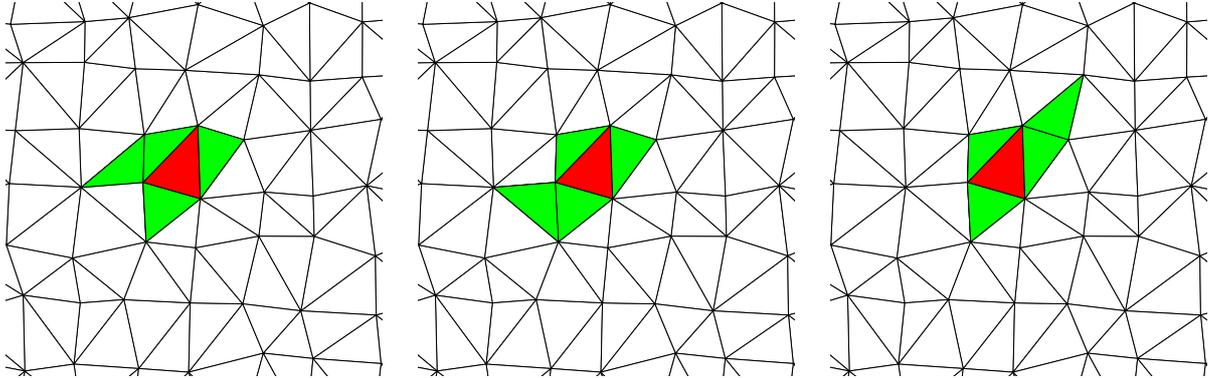


Figure 3.4: Three centred stencils of size five for the triangle T (marked red).

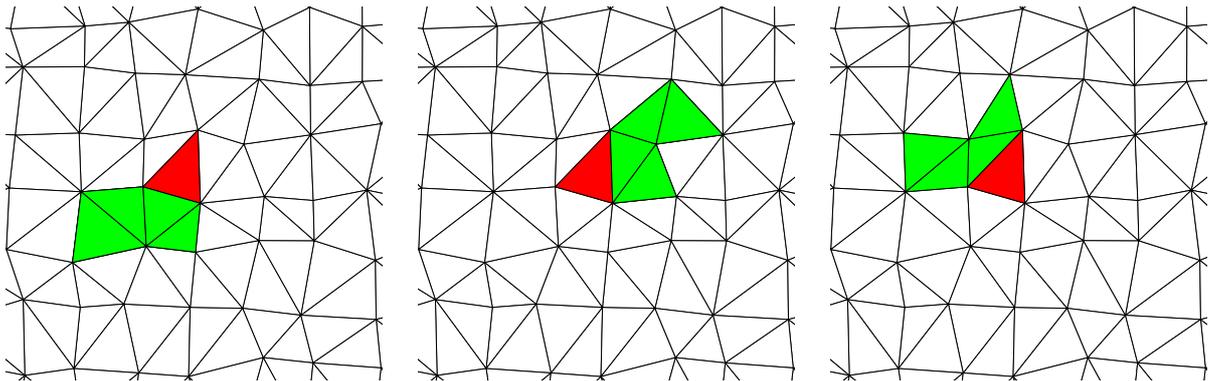


Figure 3.5: Three forward stencils of size five for the triangle T (marked red).

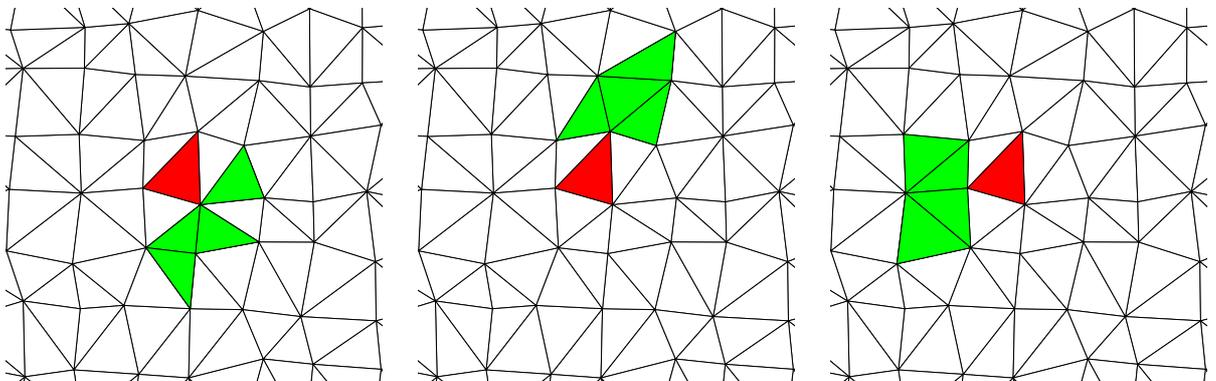


Figure 3.6: Three backward stencils of size five for the triangle T (marked red).

3.6.1 Enhanced flexibility in stencil selection for polyharmonic splines

Note that the solution of (3.22) consists of $n + q$ conditions, where we merely require $n > q$ for the well-posedness of the reconstruction problem (3.20), (3.21). But otherwise, there is no further restriction on the number $n = \#\mathcal{S}$ of given data $\{\lambda_T(u)\}_{T \in \mathcal{S}}$. This makes up a main difference to polynomial reconstruction, where we require $n = q$, in which case the number of given data (i.e. number of cell averages) is for any individual stencil dictated by the chosen degree of the polynomial space. In fact, this restriction is considered as a major drawback of the polynomial reconstruction scheme as it may be hard to find the exact degrees of freedom in unstructured grids. In contrast, the polyharmonic spline reconstruction scheme is much less restrictive when it comes to the selection of the individual stencils and their sizes. Indeed, the additional freedom allows for more flexible construction strategies for the stencil selection.

After extensive numerical tests, we observed that for reconstruction with thin plate splines, seven stencils of size four were most suitable and gave very good approximations for problems where the solution was smooth everywhere at all times. This required us to solve a 7×7 linear system for each stencil. On the other hand, when we treated problems with discontinuities or steep slopes in the solution, seven stencils of size seven were preferred because we were able to get stencils where the interpolant was non-oscillatory. The seven stencils used in both cases consisted of one centered stencil, three stencils in the forward sector and three backward stencils. For $\phi_{2,3}(r) = r^4 \log(r)$, we found that using nine stencils of size nine gave a stable reconstruction in all cases.

The flexibility in stencil sizes also allows us to vary the size of the stencil used for reconstruction on different cells on the basis of the behaviour of the solution around the cell. We implement this simple stencil adaptivity strategy in Chapter 5.

3.7 The WENO Reconstruction

As mentioned in Chapter 1, the main idea of the ENO reconstruction method is to construct several candidate recovery functions s_i on the stencils \mathcal{S}_i for a cell T and to choose the one with the least oscillation. The level of oscillation is computed using some indicator that assesses the smoothness of s_i . However, as was observed in [9, 38], this free adaptation of stencils is definitely not necessary in smooth regions and may cause loss of accuracy. In addition, the stencils selected cover a large number of cells and using one stencil means that only a limited number of cells (i.e. limited amount of information) are used in forming the reconstruction. If all the cells are used, it may be possible to obtain greater accuracy in smooth regions since all the relevant information

is used. In one space dimension, the ENO stencil selection procedure also involves many logical 'if' structures which may not always be efficient to implement.

The WENO technique is an attempt to improve on ENO with regards to the points mentioned above. In the WENO reconstruction method, instead of digitally selecting the least oscillatory recovery function we use a weighted sum:

$$s := \sum_i \omega_i s_i$$

of the recovery functions. The positive weights ω_i with $\sum_i \omega_i = 1$ are chosen in such a way that ω_i is small if the oscillation of s_i is high, reflecting the fact that the stencil lies in a region where the solution is subject to strong variation, and ω_i is larger for less oscillating s_i , i.e. the stencil \mathcal{S}_i lies in regions where the solution is smooth.

The method can be summarised in the algorithm below:

Algorithm 3.1 (WENO Reconstruction)

Input: triangle $T \in \mathcal{T}$, stencils $\mathcal{S} = \{\mathcal{S}_i\}_i$ satisfying $T \in \mathcal{S}_i \subset \mathcal{T}$ for all i .

(1) FOR each stencil \mathcal{S}_i **DO**

(1a) Compute reconstruction s_i from $\{\lambda_T(u)\}_{T \in \mathcal{S}_i}$ satisfying $s|_{\mathcal{S}_i} = u|_{\mathcal{S}_i}$;

(1b) Compute oscillation $\mathcal{I}(s_i)$ of s_i according to oscillation indicator \mathcal{I} ;

(2) Compute non-negative weights ω_i satisfying $\sum_i \omega_i = 1$ from values $\mathcal{I}(s_i)$.

Output: WENO reconstruction

$$s(\mathbf{x}) = \sum_i \omega_i s_i(\mathbf{x}). \quad (3.32)$$

We remark here that the WENO reconstruction method we implement in this work follows the construction of Friedrich [38] which has also been used in [10, 73, 145]. A slightly different formulation was proposed by Hu & Shu [59] who computed a set of so-called optimal linear weights. These linear weights, which are embedded in the WENO weights, are obtained by expressing a higher degree polynomial as a convex combination of lower degree polynomials. With this technique, they were able to get an extra order of convergence using the RK-WENO method on triangular meshes. The WENO reconstruction we use only provides better accuracy over the corresponding ENO schemes using the same stencils. We also observed some extra convergence for the linear advection equation due to the improved accuracy of the WENO reconstruction method. However, our primary aim is to show the suitability and advantages of using polyharmonic splines in the WENO framework.

3.7.1 The oscillation indicator and weights

The WENO reconstruction scheme requires an oscillation indicator \mathcal{I} which measures for any stencil \mathcal{S}_i the smoothness of the corresponding reconstruction s_i . To this end, the *Beppo Levi spaces* $\text{BL}_k(\mathbb{R}^d)$, the optimal recovery spaces for polyharmonic splines, defines the oscillation indicator \mathcal{I} as

$$\mathcal{I}(s) = |s|_{\text{BL}_k(\mathbb{R}^d)}^2, \quad \text{for } s \in \text{BL}_k(\mathbb{R}^d).$$

Following Powell [95], the semi-norm $|s|_{\text{BL}_k(\mathbb{R}^d)}^2$ can be computed simply by using the following formula

$$|s|_{\text{BL}_k(\mathbb{R}^d)}^2 = \mathbf{c}^T A \mathbf{c}, \quad (3.33)$$

where \mathbf{c} and A are defined in (3.22). When using the stable evaluation of s^h given in (3.31), the vector \mathbf{c} in (3.33) will consist of the first n elements of the vector α where

$$\mathbf{A}_1 \alpha = u_h.$$

In addition, A will be the matrix A_1 in \mathbf{A}_1 . The ease of computing the oscillation indicator using (3.33) makes the WENO reconstruction with polyharmonic splines particularly attractive.

Thus, for each triangle $T \in \mathcal{T}$ we use the oscillation indicator \mathcal{I} to compute for any polyharmonic spline reconstruction s_i its corresponding weight ω_i . We first of all compute the intermediate values

$$\tilde{\omega}_i = \frac{1}{(\epsilon + \mathcal{I}(s_i))^\rho} \quad \text{for some } \epsilon, \rho > 0, \quad (3.34)$$

where the parameter ϵ is chosen to avoid division by zero, and ρ is a measure of the sensitivity of the weights to the oscillation indicator. In our numerical experiments, following [59], we set $\epsilon = 10^{-6}$ and $\rho = 2$. The computations are generally not very sensitive to the values of these parameters.

The non-negative weights for the polyharmonic spline WENO reconstruction s in (3.32) are then, for any i , given by

$$\omega_i = \frac{\tilde{\omega}_i}{\sum_j \tilde{\omega}_j}.$$

Note that the weights ω_i form a partition of unity, i.e. $\sum_i \omega_i = 1$.

For any reference triangle $T \in \mathcal{T}$, the resulting approximation $s \equiv s_T$ to u over T is used to replace u in the numerical flux (3.14), where in particular u_{in} is replaced by s_{in} and u_{out} is replaced by s_{out} .

3.8 Numerical Experiments

3.8.1 Numerical implementation

Let Ω^h be a computational domain with a conforming triangulation \mathcal{T}^h defined on it. Let \mathcal{S}^h be a stencil from the triangulation \mathcal{T}^h for a triangle T^h and let \mathbf{x}_0 be the barycentre of $T^h \in \mathcal{S}^h \subset \mathcal{T}^h$. If $\mathbf{x}_0 + h\mathbf{x} \in \mathcal{S}^h$, then $\mathbf{x}_0 + \mathbf{x} \in \mathcal{S}^1$ where we take h to be the diameter of the inscribed circle of T^h . More precisely, we define $T^1 \in \mathcal{S}^1$ as

$$T^1 = \{\mathbf{x}_0 + \mathbf{x} : \mathbf{x}_0 + h\mathbf{x} \in T^h, \mathbf{x}_0 \text{ is the barycentre of } T^h \in \mathcal{S}^h\}$$

and \mathbf{x}_0 is also the barycentre of T^1 . The stencil \mathcal{S}^1 can therefore be defined for T^1 by scaling all the triangles $T^h \in \mathcal{S}^h$. We use the scaled stencil, the stability result in Section 3.5 and numerical quadrature to solve the reconstruction problem (3.25).

3.8.2 Linear advection

In this section we solve the two-dimensional linear advection equation

$$u_t + u_{x_1} + u_{x_2} = 0, \quad \text{for } u \equiv u(t, \mathbf{x}) \quad \text{with} \quad \mathbf{x} = (x_1, x_2) \in \mathbb{R}^2, \quad (3.35)$$

with the following initial condition [84]

$$u_0(\mathbf{x}) = u(0, \mathbf{x}) = \sin^2\left(\pi\left(x_1 + \frac{1}{2}\right)\right) \sin^2\left(\pi\left(x_2 + \frac{1}{2}\right)\right) \quad (3.36)$$

on the computational domain $\Omega = [-0.5, 0.5] \times [-0.5, 0.5] \subset \mathbb{R}^2$, by using our proposed RK-WENO method. We use periodic boundary conditions and carry out the computation for the time interval $I = [0, 1]$ so that the reference solution $\hat{u}(t, \mathbf{x})$ coincides at final time $t = 1$ with the initial condition u_0 in (3.36), so that $\hat{u}(t, \mathbf{x}) \equiv u_0(\mathbf{x})$. The velocity field and initial condition for this problem are shown in Figure 3.7.

For the WENO reconstruction, we use $\phi_1(r) = r$, the thin plate spline $\phi_{2,2}(r) = r^2 \log(r)$ and $\phi_{2,3}(r) = r^4 \log(r)$ and we perform our numerical experiments on a sequence of triangular meshes of sizes $h = \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}$. The meshes for $h = \frac{1}{8}$ to $\frac{1}{64}$ are shown in Figure 3.8. We use a CFL coefficient $c = 0.5$ in all computations. We remark that although $\phi_1(r) = r$ is not a polyharmonic spline in \mathbb{R}^2 , it is a conditionally positive definite function and thus is associated with a native Hilbert space in which it solves an optimal recovery problem. We are interested $\phi_1(r) = r$ because it is an RBF of low order which will be utilised in the implementation of mesh & order adaptivity in Chapter 6.

For the time stepping, we use SSPRK(2,2) for the WENO reconstruction with $\phi_1(r) = r$ while for $\phi_{2,2}(r) = r^2 \log(r)$ and $\phi_{2,3}(r) = r^4 \log(r)$ we use SSPRK(3,3).

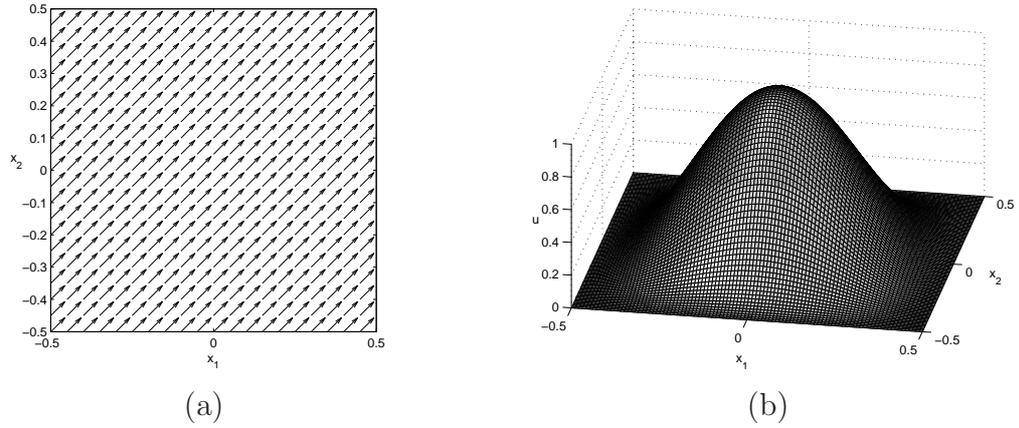


Figure 3.7: (a) Velocity field and (b) initial condition for linear advection equation.

h	$E_1(h)$	k_1	$E_2(h)$	k_2	$E_\infty(h)$	k_∞
1/8	$1.6482 \cdot 10^{-1}$	—	$2.0307 \cdot 10^{-1}$	—	$5.4104 \cdot 10^{-1}$	—
1/16	$8.5745 \cdot 10^{-2}$	0.94	$1.0664 \cdot 10^{-1}$	0.93	$3.0339 \cdot 10^{-1}$	0.83
1/32	$4.6552 \cdot 10^{-2}$	0.88	$5.8503 \cdot 10^{-2}$	0.86	$1.7330 \cdot 10^{-1}$	0.81
1/64	$2.3901 \cdot 10^{-2}$	0.97	$3.0283 \cdot 10^{-2}$	0.95	$8.8660 \cdot 10^{-2}$	0.97
1/128	$1.1870 \cdot 10^{-2}$	1.01	$1.5675 \cdot 10^{-2}$	1.02	$4.4638 \cdot 10^{-2}$	0.99

Table 3.1: Linear advection. Results by RK-WENO method using $\phi_1(r) = r$.

h	$E_1(h)$	k_1	$E_2(h)$	k_2	$E_\infty(h)$	k_∞
1/8	$6.8217 \cdot 10^{-2}$	—	$8.8569 \cdot 10^{-2}$	—	$2.5821 \cdot 10^{-1}$	—
1/16	$1.2539 \cdot 10^{-2}$	2.44	$1.5934 \cdot 10^{-2}$	2.47	$4.7370 \cdot 10^{-2}$	2.44
1/32	$2.9192 \cdot 10^{-3}$	2.10	$3.9512 \cdot 10^{-3}$	2.01	$1.4845 \cdot 10^{-2}$	1.68
1/64	$6.1218 \cdot 10^{-4}$	2.25	$8.4849 \cdot 10^{-4}$	2.22	$3.5419 \cdot 10^{-3}$	2.07
1/128	$1.2348 \cdot 10^{-4}$	2.31	$1.7230 \cdot 10^{-4}$	2.30	$7.4449 \cdot 10^{-4}$	2.21

Table 3.2: Linear advection. Results by RK-WENO method using $\phi_{2,2}(r) = r^2 \log(r)$.

h	$E_1(h)$	k_1	$E_2(h)$	k_2	$E_\infty(h)$	k_∞
1/8	$6.1235 \cdot 10^{-2}$	—	$7.8624 \cdot 10^{-2}$	—	$2.3586 \cdot 10^{-1}$	—
1/16	$6.8985 \cdot 10^{-3}$	3.15	$8.7355 \cdot 10^{-3}$	3.17	$3.0102 \cdot 10^{-2}$	2.97
1/32	$9.1148 \cdot 10^{-3}$	2.92	$1.2630 \cdot 10^{-3}$	2.79	$4.6647 \cdot 10^{-3}$	2.69
1/64	$1.1159 \cdot 10^{-4}$	3.03	$1.5040 \cdot 10^{-4}$	3.07	$6.7914 \cdot 10^{-4}$	2.78
1/128	$1.3196 \cdot 10^{-5}$	3.08	$1.8670 \cdot 10^{-5}$	3.01	$8.4306 \cdot 10^{-5}$	3.01

Table 3.3: Linear advection. Results by RK-WENO method using $\phi_{2,3}(r) = r^4 \log(r)$.

We denote the numerical solution by u_h and the errors and the corresponding con-

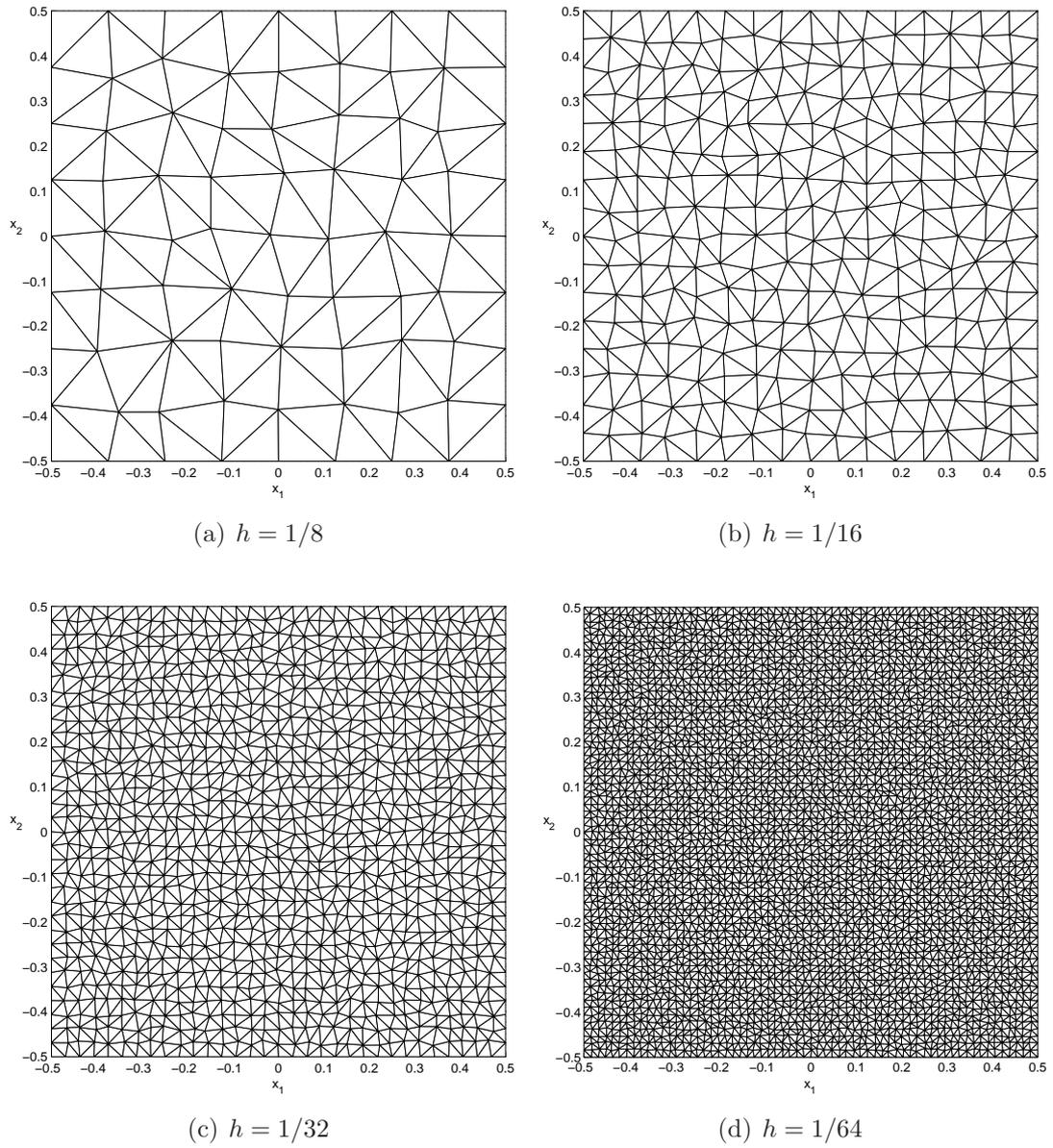


Figure 3.8: Sequence of four meshes with their mesh sizes.

vergence rates are computed as

$$E_p(h) = \|u_h - \hat{u}\|_p \quad \text{and} \quad k_p = \frac{\log[E_p(h)/E_p(h/2)]}{\log(2)}, \quad \text{for } p = 1, 2, \infty,$$

for the norms $\|\cdot\|_1$, $\|\cdot\|_2$, and $\|\cdot\|_\infty$. The results are presented in Tables 3.1 – 3.3.

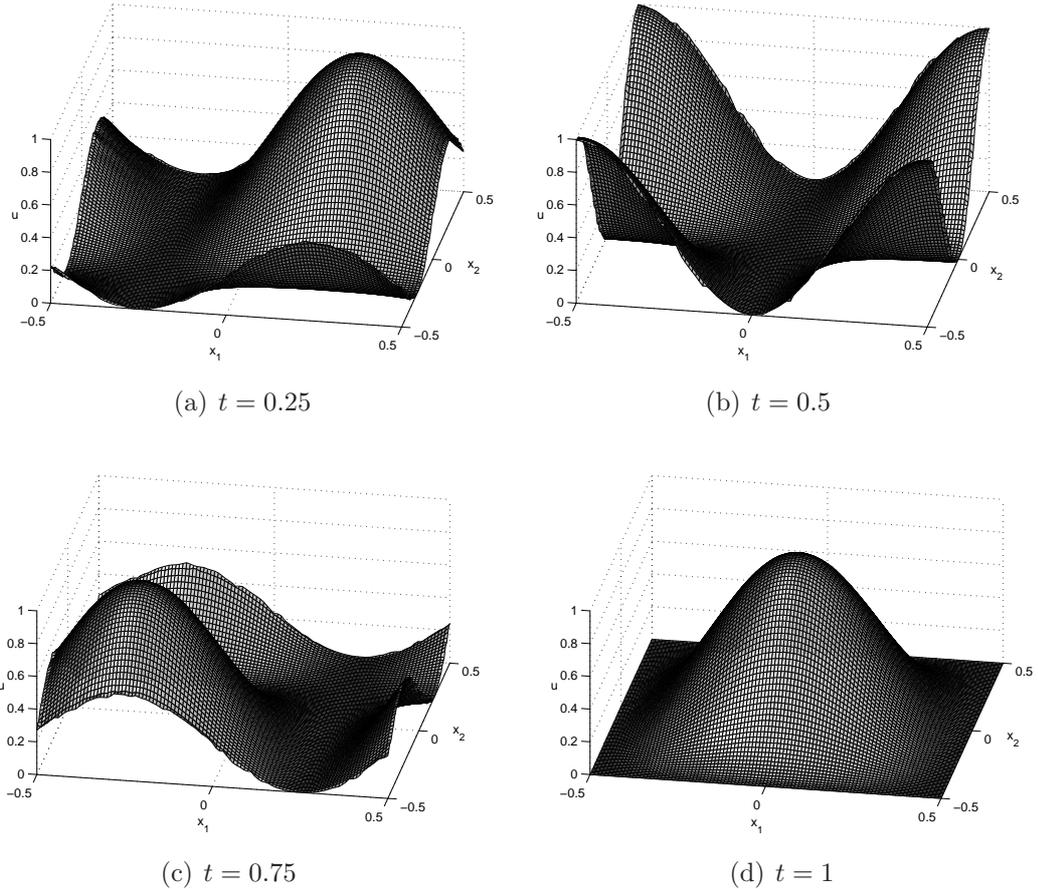


Figure 3.9: Solution of (3.35), (3.36) at times $t = 0, 0.25, 0.5$, and 1 using thin plate spline reconstruction.

The scheme with $\phi_1(r) = r$ is implemented with four stencils of size four where we use one centred stencil, one forward stencil and two backward stencils. For thin plate splines we use seven stencils of size four, including one centred stencil, three forward stencils and three backward stencils. Finally, for $\phi_{2,3}(r) = r^4 \log(r)$, we use nine stencils of size nine corresponding to three centred stencils, three forward stencils and three backward stencils. We see that the RK-WENO scheme with $\phi_1(r) = r$ yields almost first order convergence while the thin plate splines give at least second order convergence. In fact, we observe some extra convergence in this case particularly in the L_1 - and L_2 -norms. We also observe third order convergence for $\phi_{2,3}(r) = r^4 \log(r)$. This agrees with what we expect, since our WENO reconstruction scheme is based on local Lagrange

reconstruction with order $p = 2$ for $\phi_{2,2}(r) = r^2 \log(r)$ and $p = 3$ for $\phi_{2,3}(r) = r^4 \log(r)$. We also observe that the accuracy improves as the order of reconstruction increases.

3.8.3 Burgers' equation

We further test the accuracy of the RK-WENO method on the two-dimensional nonlinear Burgers' equation

$$u_t + \left(\frac{1}{2}u^2\right)_{x_1} + \left(\frac{1}{2}u^2\right)_{x_2} = 0, \quad (3.37)$$

with the following initial condition

$$u_0(\mathbf{x}) = u(0, \mathbf{x}) = \frac{1}{4} + \frac{1}{2} \sin(\pi(x_1 + x_2)), \quad (3.38)$$

on the computational domain $\Omega = [-1, 1] \times [-1, 1] \subset \mathbb{R}^2$, with periodic boundary conditions. We carry out the computations to time $t = 0.1$ where the solution is still smooth but we have to use Newton's method to solve the implicit characteristic relation

$$u = \frac{1}{4} + \frac{1}{2} \sin(\pi((x_1 - ut) + (x_2 - ut))),$$

to obtain it.

h	$E_1(h)$	k_1	$E_2(h)$	k_2	$E_\infty(h)$	k_∞
1/8	$1.2241 \cdot 10^{-1}$	—	$8.0217 \cdot 10^{-2}$	—	$1.2673 \cdot 10^{-1}$	—
1/16	$6.3361 \cdot 10^{-2}$	0.95	$4.1812 \cdot 10^{-2}$	0.94	$6.7913 \cdot 10^{-2}$	0.83
1/32	$3.2207 \cdot 10^{-2}$	0.98	$2.0638 \cdot 10^{-2}$	1.02	$3.8495 \cdot 10^{-2}$	0.81
1/64	$1.5676 \cdot 10^{-2}$	1.04	$9.9286 \cdot 10^{-3}$	1.06	$2.1822 \cdot 10^{-2}$	0.97
1/128	$7.5710 \cdot 10^{-3}$	1.05	$4.8286 \cdot 10^{-3}$	1.04	$1.0761 \cdot 10^{-2}$	1.01

Table 3.4: Burgers' equation. Results by RK-WENO method using $\phi_1(r) = r$.

h	$E_1(h)$	k_1	$E_2(h)$	k_2	$E_\infty(h)$	k_∞
1/8	$7.9604 \cdot 10^{-2}$	—	$6.2258 \cdot 10^{-2}$	—	$1.3799 \cdot 10^{-1}$	—
1/16	$1.8440 \cdot 10^{-2}$	2.11	$1.4125 \cdot 10^{-2}$	2.14	$3.5223 \cdot 10^{-2}$	1.97
1/32	$5.3053 \cdot 10^{-3}$	1.80	$4.3031 \cdot 10^{-3}$	1.72	$9.9759 \cdot 10^{-3}$	1.82
1/64	$1.3533 \cdot 10^{-3}$	1.97	$1.3173 \cdot 10^{-3}$	1.71	$3.2681 \cdot 10^{-3}$	1.61
1/128	$2.9864 \cdot 10^{-4}$	2.18	$2.9680 \cdot 10^{-4}$	2.15	$8.8788 \cdot 10^{-4}$	1.88

Table 3.5: Burgers' equation. Results by RK-WENO method using $\phi_{2,2}(r) = r^2 \log(r)$.

The results are shown in Tables 3.4 – 3.6. Once again we observe the expected order of convergence.

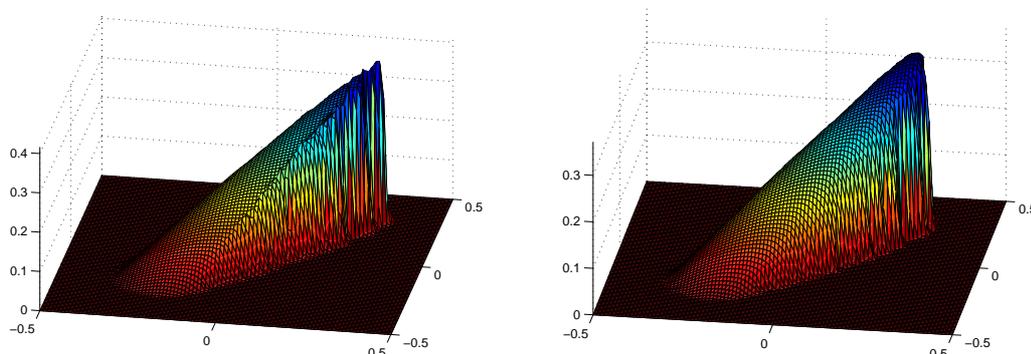
h	$E_1(h)$	k_1	$E_2(h)$	k_2	$E_\infty(h)$	k_∞
1/8	$5.9812 \cdot 10^{-2}$	—	$4.3381 \cdot 10^{-2}$	—	$1.7325 \cdot 10^{-1}$	—
1/16	$8.8297 \cdot 10^{-3}$	2.76	$6.5841 \cdot 10^{-3}$	2.72	$2.8771 \cdot 10^{-2}$	2.59
1/32	$1.3035 \cdot 10^{-3}$	3.01	$8.0051 \cdot 10^{-4}$	3.04	$3.5720 \cdot 10^{-3}$	3.01
1/64	$1.6407 \cdot 10^{-4}$	2.99	$1.0217 \cdot 10^{-4}$	2.97	$4.9199 \cdot 10^{-4}$	2.86
1/128	$1.9948 \cdot 10^{-5}$	3.04	$1.2595 \cdot 10^{-5}$	3.02	$6.5458 \cdot 10^{-5}$	2.91

Table 3.6: Burgers' equation. Results by RK-WENO method using $\phi_{2,3}(r) = r^4 \log(r)$.

We further studied the Burgers' equation with the initial condition [39]

$$u_0(\mathbf{x}) = \begin{cases} \exp\left(\frac{\|\mathbf{x}-\mathbf{c}\|^2}{\|\mathbf{x}-\mathbf{c}\|^2-R^2}\right), & \|\mathbf{x}-\mathbf{c}\| < R; \\ 0, & \text{otherwise,} \end{cases} \quad (3.39)$$

with $R = 0.15$, $\mathbf{c} = (-0.2, -0.2)^T$ on the computational domain $\Omega = [-0.5, 0.5] \times [-0.5, 0.5] \subset \mathbb{R}^2$. We solved this problem using the RK-WENO method with thin plate spline reconstruction. This was done with stencils of various sizes on a mesh with width $\frac{1}{64}$.



(a) Solution with four stencils of size four (b) Solution with seven stencils of size seven

Figure 3.10: Solution of (3.37), (3.39) on stencils of size four and seven at time $t = 1.49514$ using the RK-WENO method with thin plate spline reconstruction.

We observed the presence of unwanted oscillations in the solution when we used stencils of size four as in Figure 3.10(a). The oscillations reduced when we use stencils of size five and six but were completely absent when we used stencils of size seven, see Figure 3.10(b).

3.9 Treatment of Boundary Conditions

Boundary conditions are required in order to apply the finite volume method to update the boundary cells of a triangulation to the next time level. The proper treatment of boundary conditions is usually an essential ingredient in the design of efficient numerical schemes for conservation laws.

The finite volume method has the advantage over other methods in that it allows boundary conditions to be applied noninvasively. This is because the values of the variables are located within the control volume, and not at nodes.

For problems with periodic boundary conditions, or problems with compact support for the entire simulation, the implementation of the boundary conditions requires simply setting as many ghost cells (also known as fictitious cells) as needed, using either the periodicity condition or the compactness of the solution. The number of required ghost cells would usually depend on the stencil size [10]. Furthermore, when implementing periodic boundary conditions on unstructured grids, the same number of cell edges must occur on opposite sides of the computational domain [72].

There are several other types of boundary conditions which are all treated differently. For reflective or symmetric boundary conditions, one would set as many ghost cells as needed and then use the symmetry/antisymmetry properties to prescribe cell averages at those ghost cells [10]. For inflow boundary conditions on unstructured meshes, we use the exact physical inflow boundary condition at the Gaussian points on the boundary, i.e. for the boundary condition

$$u(t, \varrho) = g(t, \varrho), \quad \varrho \text{ is the boundary of the computational domain,}$$

we evaluate the function g at the Gaussian integration points along the inflow boundary ϱ and compute the numerical flux as

$$\mathcal{F}(u(t, G_\ell)) = F(g(t, G_\ell)).$$

Outflow or transmissive boundaries (also known as open-end, transmission, far-field or nonreflecting boundaries) may occur when implementing finite volume methods on finite computational domains. This type of boundary condition is regarded as a numerical attempt to create boundaries that allow the passage of waves without any effect on them. In the one-dimensional case, techniques for treating outflow boundaries have been well developed. However, in multidimensions, this is still an area of ongoing research. Following Toro [128] and Käser [72], we first look for all the Gaussian integration points G_ℓ on the outflow boundary and set $u_{\text{in}} = u_{\text{out}}$ where u_{out} is the boundary extrapolated value from the outside of the cell and u_{in} is the boundary extrapolated value from the

inside of the cell. We then use this to compute the Lax-Friedrich flux (3.14); this does not affect the flow inside the computational domain.

To illustrate the implementation of inflow and outflow boundary conditions, we consider the problem [94]

$$u_t + x_1 u_{x_1} - x_2 u_{x_2} = 0; \quad 1 \leq x_1, x_2 \leq 2, \quad t \geq 0, \quad (3.40)$$

in which the velocity field is divergence free. On the two inflow boundaries we have the following boundary conditions

$$\begin{aligned} u(1, x_2, t) &= 1 + x_2^2, & 1 \leq x_2 \leq 2, \quad t \geq 0, \\ u(x_1, 2, t) &= 1 + 4x_1^2, & 1 \leq x_1 \leq 2, \quad t \geq 0. \end{aligned} \quad (3.41)$$

The initial condition is

$$u(0, \mathbf{x}) = 0, \quad 1 < x_1 \leq 2, \quad 1 \leq x_2 < 2. \quad (3.42)$$

The steady state solution of this problem is

$$u(\mathbf{x}) = 1 + (x_1 x_2)^2. \quad (3.43)$$

The RK-WENO method is terminated when

$$\|u^{n+1} - u^n\|_\infty \leq 10^{-5} \quad (3.44)$$

where u^n is the RK-WENO solution at time $t = t^n$.

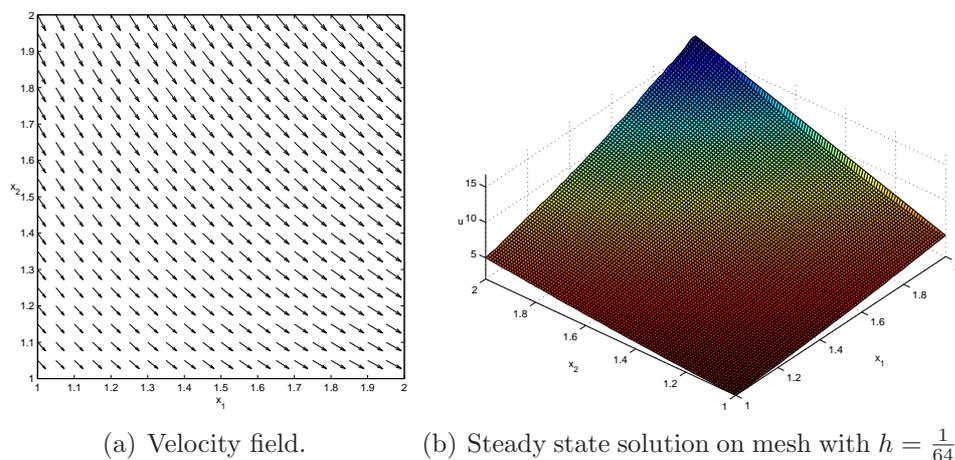


Figure 3.11: (a) Velocity field and (b) steady state solution for (3.40) - (3.42).

The results are presented in Table 3.7 and show the expected second order convergence.

h	$E_1(h)$	k_1	$E_2(h)$	k_2	$E_\infty(h)$	k_∞
1/8	$3.3589 \cdot 10^{-2}$	—	$4.3030 \cdot 10^{-2}$	—	$1.1516 \cdot 10^{-1}$	—
1/16	$8.6934 \cdot 10^{-3}$	1.95	$1.1137 \cdot 10^{-2}$	1.95	$4.1284 \cdot 10^{-2}$	1.48
1/32	$1.6471 \cdot 10^{-3}$	2.40	$2.2458 \cdot 10^{-3}$	2.31	$1.1531 \cdot 10^{-2}$	1.84
1/64	$3.0564 \cdot 10^{-4}$	2.43	$4.2257 \cdot 10^{-4}$	2.41	$2.6712 \cdot 10^{-3}$	2.10

Table 3.7: Solution of (3.40) - (3.42) by the RK-WENO method using thin plate spline reconstruction.

3.10 Doswell's Frontogenesis

We have so far shown the performance of our RK-WENO method for standard test problems and we now proceed to use the method to solve a more challenging numerical test. In this section, we present a simulation of Doswell's frontogenesis problem [33]. His idealized kinematic frontogenesis models the interaction of a nondivergent vortex with a frontal zone that was initially straight. It is now a popular but challenging test case for advection schemes as it is able to test numerically the ability of a scheme to treat discontinuities that move with respect to each other, see [55, 122].

We solve the linear equation

$$u_t + \sigma_1(\mathbf{x})u_{x_1} + \sigma_2(\mathbf{x})u_{x_2} = 0, \quad (3.45)$$

where the velocity field is a steady circular vortex with tangential velocity

$$v_t(r) = \frac{1}{v_{\max}} \cdot \frac{\tanh(r)}{\cosh^2(r)}. \quad (3.46)$$

This means that

$$\sigma_1(\mathbf{x}) = -\frac{x_2 - c_1}{r}v_t(r) \quad \text{and} \quad \sigma_2(\mathbf{x}) = \frac{x_1 - c_2}{r}v_t(r),$$

where (c_1, c_2) is the center of the rotation and r is the distance of any point in the domain from the center of rotation i.e. $r = \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2}$. The variable angular velocity is given as $\omega = v_t/r$.

In this test case, the initial condition is defined on a square domain $[-5, 5]^2$ as

$$u(0, \mathbf{x}) = u_0(x_2) = \tanh\left(\frac{x_2 - c_2}{\delta}\right), \quad (3.47)$$

where δ expresses the characteristic width of the front zone. The exact solution is given as

$$u(t, \mathbf{x}) = \tanh\left(\frac{x_2 - c_2}{\delta} \cos(\omega t) - \frac{x_1 - c_1}{\delta} \sin(\omega t)\right). \quad (3.48)$$

This solution is a spiral-like structure that starts at the vortex center.

We solve the problem for $\delta = 10^{-6}$ and $v_{\max} = 0.3849001$ with the vortex centered at the origin, i.e. $(c_1, c_2) = (0, 0)$, using the RK-WENO method with thin plate spline reconstruction. Figure 3.12 shows the velocity field and initial condition for Doswell's frontogenesis problem. The method was implemented on a specially created fixed mesh (Figure 3.13) and also on two other meshes of widths $h = \frac{1}{40}$ (Figure 3.14(a)) and $h = \frac{1}{80}$ (Figure 3.14(b)).

We first of all performed the WENO reconstruction with seven stencils of size seven on the specially created mesh. The simulation was run for time $t = 4$ and we observe that the solution becomes increasingly multi-scaled in time, with small scale features near the origin where the vortex flow is strongest. Our computational mesh in Figure 3.13 is increasing refined towards the origin due to the fact that the essentially non-oscillatory reconstruction we use may not be able to find a smooth stencil when a coarse mesh is used near the center of the vortex. If the simulation continues to advance in time, it may go beyond the resolution of the computational grid. We display the solution at times $t = 1$, $t = 2$, $t = 3$ and $t = 4$ in Figure 3.15. By the time $t = 4$, we notice that the solution has become a discontinuous rolling surface.

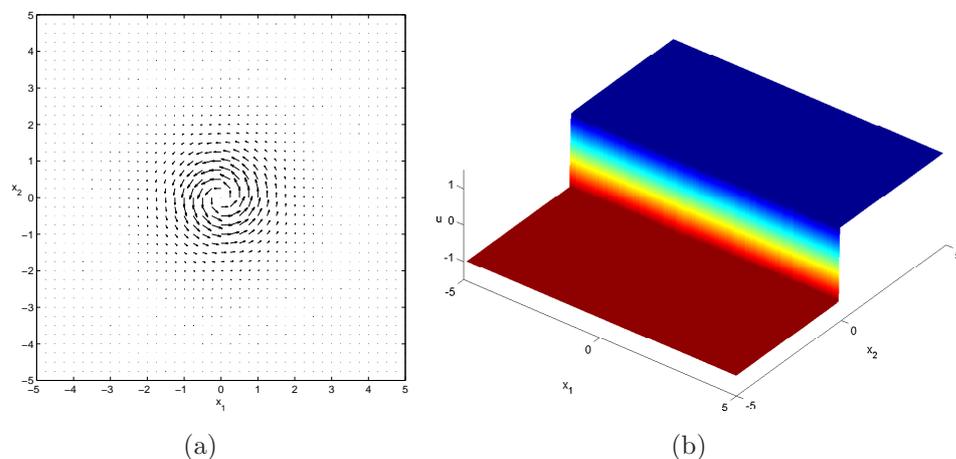


Figure 3.12: (a) Velocity field and (b) initial condition for Doswell's frontogenesis.

The simulation results are clearly in agreement with the analytic solution. For instance, we know that in the exact solution (3.48), the maximum deformation of the frontal zone is near the region where the wind is maximum, see Figure 3.12(a), and this is accurately reflected in our RK-WENO solution. Moreover, the solutions from our RK-WENO method are sharp and confirm the essentially non-oscillatory nature of the reconstruction technique.

We also solve this problem on the specially created mesh (Figure 3.13) where the WENO reconstruction is implemented with seven stencils of size four. We observe that

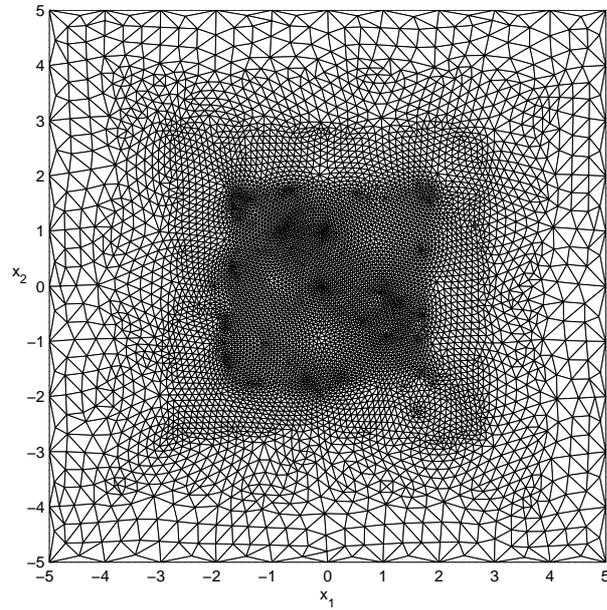


Figure 3.13: Specially created computational mesh for Doswell's frontogenesis.

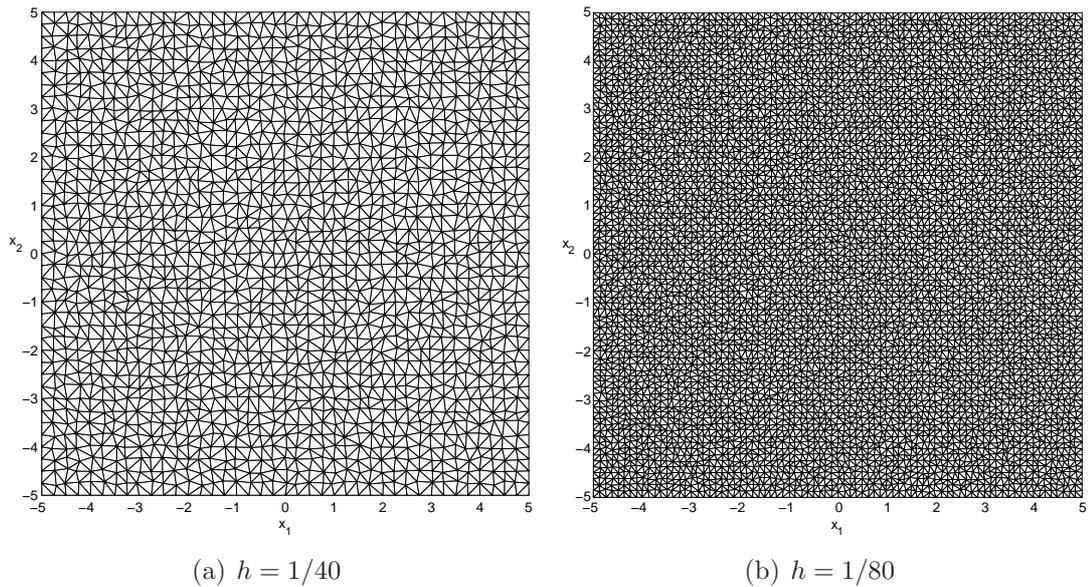


Figure 3.14: Two meshes used for the solution of Doswell's frontogenesis.

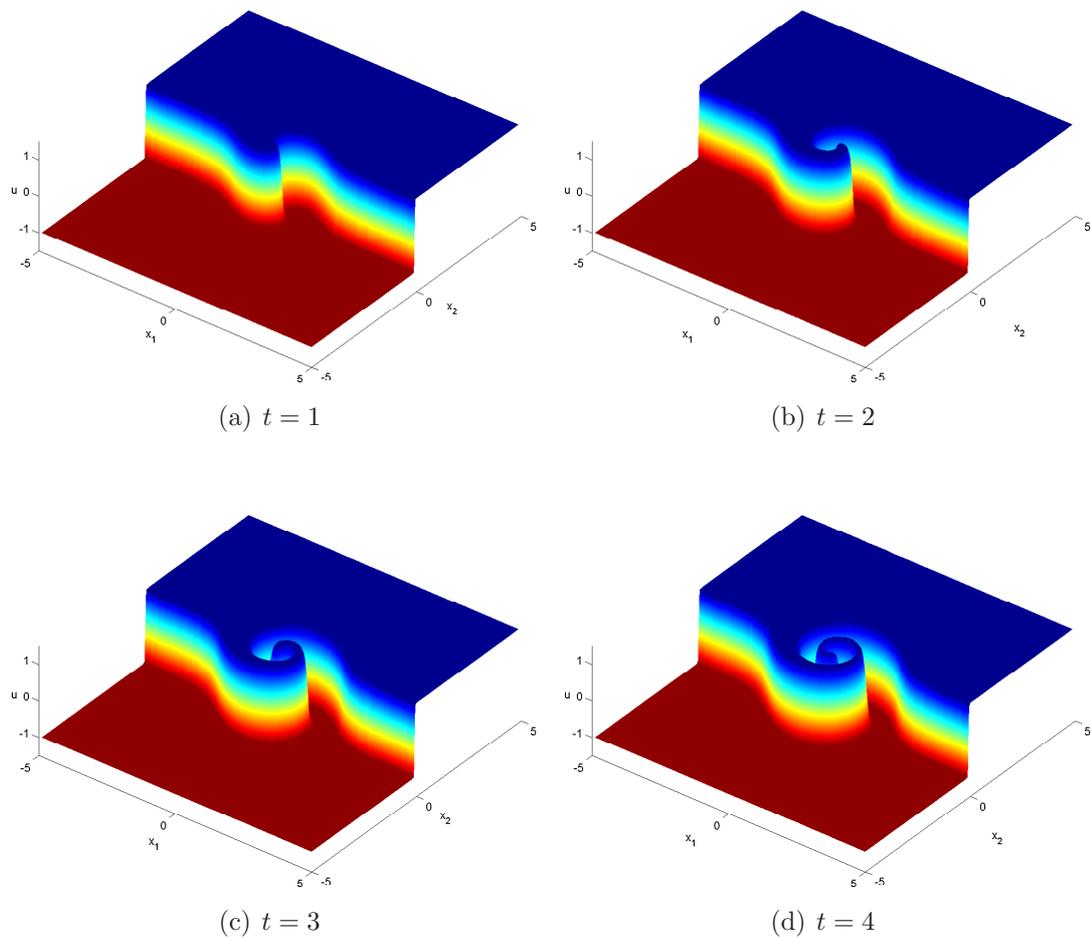
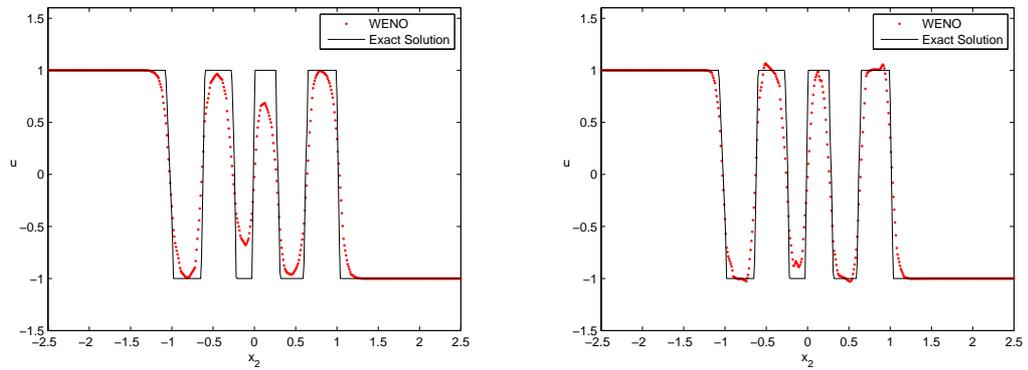
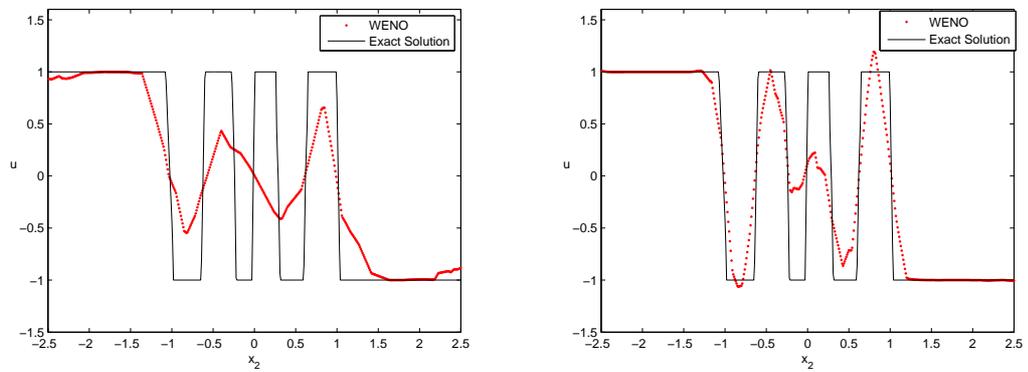


Figure 3.15: Solution of Doswell's frontogenesis problem at (a) $t = 1$; (b) $t = 2$; (c) $t = 3$; (d) $t = 4$.

Results obtained by RK-WENO method on the specially created mesh and with seven stencils of size seven.



(a) Cut for solution computed on specially created mesh with stencils of size seven. (b) Cut for solution computed on specially created mesh with stencils of size four.



(c) Cut for solution on mesh $h = 1/40$. (d) Cut for solution on mesh $h = 1/80$.

Figure 3.16: Cuts along the x_2 -axis at time $t = 4$ for different solutions .

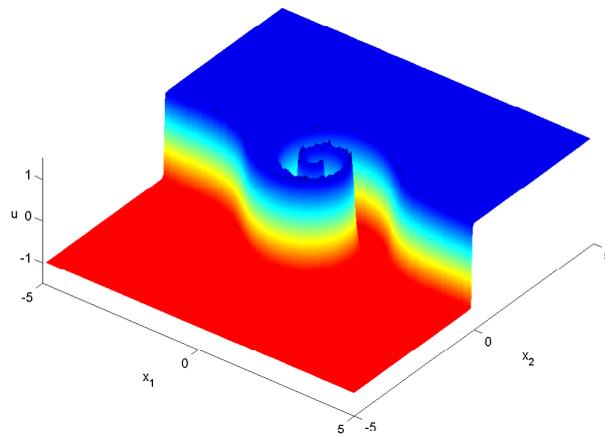


Figure 3.17: Solution of Doswell's frontogenesis problem by RK-WENO method on specially created mesh with stencils of size four at $t = 4$.

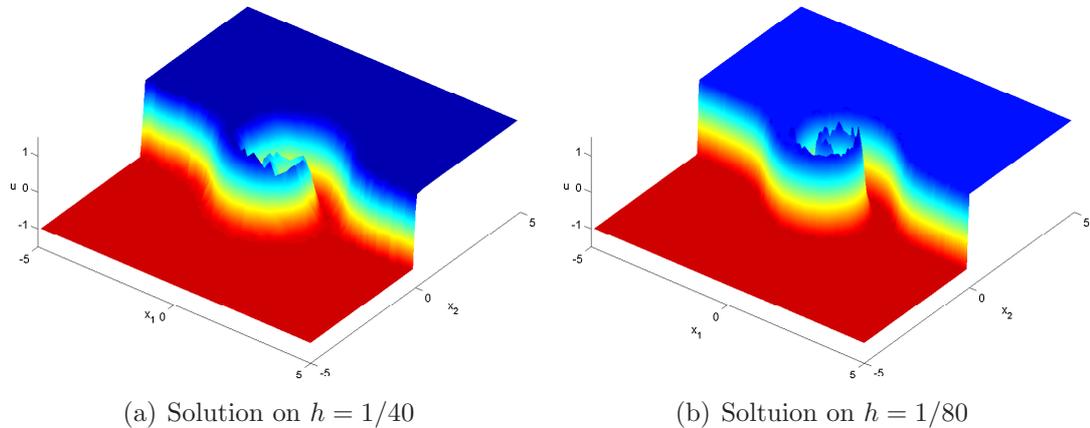


Figure 3.18: Solution of Doswell's frontogenesis at $t = 4$ on meshes with widths (a) $h = 1/40$ and (b) $h = 1/80$.

although there is little numerical diffusion, there are some oscillations near the vortex. The solution for this instance at time $t = 4$ is shown in Figure 3.17. In fact, when using the stencils of size seven there are virtually no oscillations while when using the stencils of size four we have overshoots of up to 0.260808 and undershoots of up to 0.302022 at time $t = 4$. This again confirms that for problems with steep gradients or discontinuities it is preferable to use seven stencils of size seven for thin plate spline reconstruction.

Finally, we solve the problem on two uniform meshes of widths $h = \frac{1}{40}$ and $h = \frac{1}{80}$ with the WENO reconstruction done with stencils of size seven. The mesh of width $h = \frac{1}{40}$ yields poor results with great numerical diffusion as seen in Figure 3.18(a). This is because of the inability of the mesh to resolve the fine scales of the problem as the solution advances in time. Furthermore, the solution on the mesh with size $h = \frac{1}{80}$ has less diffusion but still produces overshoots of up to 0.354603 and undershoots of up to 0.194495 at $t = 4$. The results at time $t = 4$ is shown in Figure 3.18(b). This mesh has 12800 triangles while our specially created mesh contains 13176 triangles. We therefore observe that although there is no great difference in the number of triangles, the mesh that is increasingly refined towards the origin produces far superior results. Nevertheless, we observe in Figures 3.18(a) and 3.18(b) that even when the solution is not well resolved, the method remains stable, i.e. the oscillations do not get out of hand.

Further illustration is provided in Figure 3.16 where we show one-dimensional cuts through the solution at time $t = 4$ along the x_2 -axis ($x_1 = 0$) for $-2.5 \leq x_2 \leq 2.5$. The solid lines correspond to the exact solution while the symbol '.' corresponds to the numerical solution obtained from the reconstruction with stencils of size seven (Figure 3.16(a)) and four (Figure 3.16(b)) on our specially created mesh as well as the solution obtained on the meshes of widths $h = \frac{1}{40}$ (Figure 3.16(c)) and $h = \frac{1}{80}$ (Figure 3.16(d)). These cuts show the essentially non-oscillatory nature of the method when the reconstruction

was done with seven stencils of size seven although there is slight numerical diffusion.

Chapter 4

The ADER-WENO method

There are currently several numerical schemes for hyperbolic conservation laws that yield high order in space. These include the ENO & RK-WENO schemes described in the previous chapters, spectral methods [10] and the discontinuous Galerkin finite element methods [23]. Although all these schemes are of high order in space, they are typically discretised in time with Runge-Kutta methods. To avoid unwanted oscillations, the Runge-Kutta methods usually need to be SSP (see Section 3.3). However, Ruuth & Spiteri [98] have shown that the accuracy of the SSP Runge-Kutta methods is essentially limited, and may consequently place a restriction on the order of the entire scheme. It has also been proven that all explicit Runge-Kutta methods of order higher than four need more stages than their order, i.e. $s > p$ [44]. Moreover, as noted in Chapter 3, fourth and fifth order SSP methods are quite complicated, require extra operator computations and have a reduced region of absolute stability. Thus, in many practical applications, a third order SSP Runge-Kutta method is used even for methods with higher order discretisations in space.

The ADER-WENO method is a recent Godunov-type non-oscillatory finite volume scheme for hyperbolic conservation laws which can be viewed as a generalization of the classical first order Godunov method to arbitrary high orders. It was designed to yield arbitrary high order of accuracy in both space and time for smooth solutions by combining high order WENO polynomial reconstruction from cell averages with high order time integration of the numerical flux using the Cauchy-Kowalewski procedure and the solution of generalized Riemann problems.

The ADER-WENO method was first developed in 2001 for linear advection problems with constant coefficients by Toro *et al* [129, 130]. It was initially designed as an extension of the Modified Generalized Riemann Problem of Toro [128], which is a simplification of the Generalized Riemann Problem (GRP)-type scheme of Ben-Artzi and Falcovitz [13]. The successful extension of the ADER-WENO approach to non-linear problems, systems of conservation laws and other convection dominated problems

relies on the solution of the Generalized Riemann Problem [131, 133]. Titarev and Toro [125, 127, 132] extended the ADER-WENO schemes to the solution of scalar non-linear problems while Schwartzkopff *et al* [105] applied the method to two-dimensional systems of conservation laws on Cartesian grids. Furthermore, Käser and Iske [73] developed a version of ADER-WENO finite volume schemes for unstructured triangular meshes. Dumbser and Munz [35] have also used the ADER methodology for the temporal discretisation of the discontinuous Galerkin method.

In this chapter, we present an extension of the ADER-WENO schemes in order to solve conservation laws on unstructured triangulations. We propose a numerical method that combines high order local polyharmonic spline reconstruction with a high order flux evaluation method to update cell averages through fluxes across the cell interfaces. Thus, the main difference between our formulation of the ADER-WENO method and the previous ADER-WENO schemes in the literature lies in the reconstruction technique. Unlike the previous schemes that usually rely on WENO polynomial reconstruction method, we will employ the polyharmonic spline WENO reconstruction. In addition, because the flux evaluation in the ADER-WENO method is based on the solution of a set of GRPs at the cell interfaces, the initial data for the GRPs in this chapter will be the polyharmonic spline interpolant and its derivatives evaluated at the cell interfaces.

In this work, we will implement the ADER-WENO method with second and third order polyharmonic spline reconstructions only. We use these reconstructions to validate the proposed method and also as a basis for future higher order reconstructions.

4.1 The ADER-WENO Finite Volume Scheme

In this chapter, we will use the ADER-WENO finite volume method to solve the two dimensional scalar conservation law

$$\frac{\partial u}{\partial t} + \nabla \cdot F(u) = 0, \quad (4.1)$$

where $F(u) = (f_1(u), f_2(u))^T$. Integrating (4.1) over a space-time control volume $T \times [t^n, t^{n+1}]$, $T \in \mathcal{T}$, we obtain:

$$\bar{u}_T(t^{n+1}) = \bar{u}_T(t^n) - \frac{1}{|T|} \left(\sum_{j=1}^3 \int_{t^n}^{t^{n+1}} \int_{\partial T_j} F(u) \cdot \mathbf{n}_j ds dt \right), \quad (4.2)$$

where $\bar{u}_T(t^n)$ is the cell average of the solution in triangle $T \in \mathcal{T}$ at time $t = t^n$, $|\partial T_j|$, $j = 1, 2, 3$ denote the length of the three edges of the triangle and \mathbf{n}_j in (4.2) is the outer normal vector of the edge ∂T_j . We approximate (4.2) by the following one-step finite

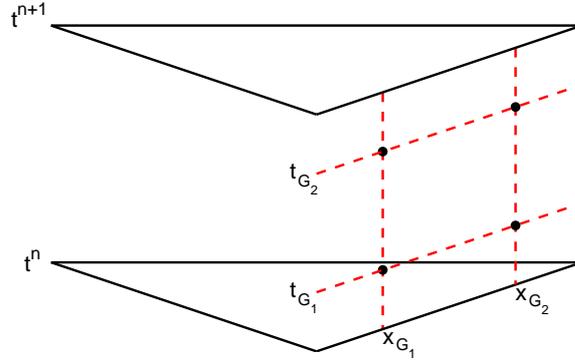


Figure 4.1: Flux evaluation across the edge of a triangle from t^n to t^{n+1} using two Gaussian integration points in space and time.

volume scheme:

$$\bar{u}_T^{n+1} = \bar{u}_T^n - \frac{\Delta t}{|T|} \sum_{j=1}^3 \tilde{F}_{T,j}^n, \quad (4.3)$$

where $\Delta t = t^{n+1} - t^n$ and \bar{u}_T^n is an approximation to $\bar{u}_T(t^n)$. The numerical flux $\tilde{F}_{T,j}^n$ across each boundary ∂T_j , $j = 1, 2, 3$, of the control volume $T \in \mathcal{T}$ during the time interval $[t^n, t^{n+1}]$ is an approximation to the time-average of the physical flux:

$$\hat{F}_{T,j} = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \left(\int_{\partial T_j} F(u) \cdot \mathbf{n}_j ds \right) dt. \quad (4.4)$$

Thus, if we discretise the spatial integral over the face of the control volumes and the time integral in (4.4) using suitable Gaussian numerical quadrature, the numerical flux will be given by

$$\tilde{F}_{T,j}^n = \sum_{\alpha=1}^{q_t} K_\alpha |\partial T_j| \sum_{\beta=1}^{q_x} L_\beta F(u(t_{G_\alpha}, \mathbf{x}_{G_\beta})) \cdot \mathbf{n}_j, \quad (4.5)$$

where q_t and q_x are the number of integration points in time and space respectively. The weights K_α , L_β , and the integration points $(t_{G_\alpha}, \mathbf{x}_{G_\beta})$ of the time and space discretisation are determined by an appropriate Gaussian quadrature. Figure 4.1 illustrates the situation where we have two Gaussian integration points in space and time, i.e. $q_x = q_t = 2$. Since we are constructing schemes only up to third order accuracy, two Gaussian points will be sufficient for all our computations, c.f. [72, 73, 59].

According to Titarev & Toro [125], the ADER approach consists of three basic steps

- reconstruction of pointwise values of the solution and its derivatives from cell

averages at the cell interface,

- solution of the Generalized Riemann Problem (GRP) at the cell interface,
- computation of the fluxes at the cell interface to update the solution to the next time step using the conservative scheme (4.3).

There are several possible reconstruction techniques that can be employed. For instance, we can use a single stencil for reconstruction to obtain a linear ADER scheme. However, to avoid spurious oscillations near steep gradients and discontinuities, it is pertinent to use a non-oscillatory reconstruction method. To this end, we use the polyharmonic spline WENO reconstruction proposed in Chapter 3, which leads us to a nonlinear scheme.

4.1.1 The Generalized Riemann Problem at the cell interface

Let us first formulate the GRP which is the basis of the high order flux evaluation of the ADER-WENO method. Recall that based on the WENO reconstruction technique, the solution u at time $t = t^n$ is represented by a recovery function on each triangular cell $T \in \mathcal{T}$. The recovery functions can be used to define a Generalized Riemann Problem at each cell interface.

Conventionally, the Riemann problem for a conservation law is defined as a Cauchy problem with initial conditions consisting of two constant states separated by a discontinuity at the origin. A Generalized Riemann Problem, however, involves solving a Cauchy problem with piecewise smooth but not constant initial data. In the numerical solution of conservation laws, the cell interface can be treated as a local origin for a Riemann problem or GRP while the cell averages or recovery functions on both sides of the cell interface constitute the states.

The GRP was first implemented on unstructured meshes by Käser & Iske [73]. In [73], at each Gaussian point \mathbf{x}_{G_β} on a cell interface, the multidimensional problem was reduced to a series of one-dimensional GRPs oriented in the normal direction to the interface. Thus, the resulting one dimensional Cauchy problem is given as

$$u_t + \partial_{\mathbf{n}} F(u) = 0, \quad (4.6)$$

$$u(0, x_{\mathbf{n}}) = \begin{cases} s_L(x_{\mathbf{n}}), & x_{\mathbf{n}} < 0, \\ s_R(x_{\mathbf{n}}), & x_{\mathbf{n}} > 0, \end{cases} \quad (4.7)$$

where $x_{\mathbf{n}}$ is a local spatial coordinate oriented along the outer normal with the origin \mathbf{x}_{G_β} as shown in Figure 4.2 and the partial derivative $\partial_{\mathbf{n}}$ is the derivative in the direction normal to the interface. Moreover, s_L is the recovery function (polyharmonic spline

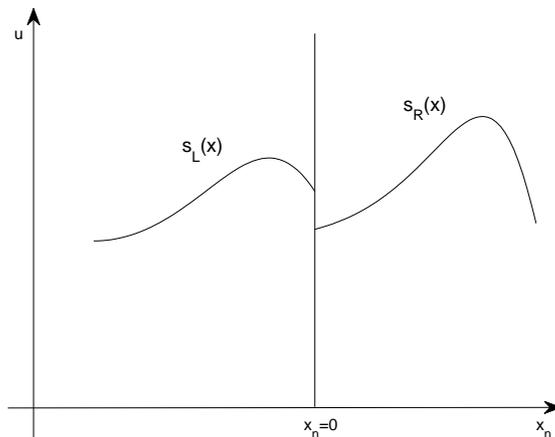


Figure 4.2: The GRP along the outward pointing unit vector with polyharmonic spline reconstructions $s_L(x)$ and $s_R(x)$ approximating the solution u .

interpolant in our case) on the actual cell and s_R is the recovery function on the adjacent cell.

4.1.2 Flux evaluation

One of the main goals of the ADER-WENO method is to compute an approximation to the function values $u(t_{G_\alpha}, \mathbf{x}_{G_\beta})$, $1 \leq \alpha \leq q_t$, $1 \leq \beta \leq q_x$ at the Gaussian integration points $(t_{G_\alpha}, \mathbf{x}_{G_\beta})$. These function values, sometimes referred to as the *states* of the solution at the cell interface, are used in the approximation of flux function F in (4.5). These approximate values of F are in turn used in evaluating the numerical flux \tilde{F} .

In the ADER methodology, an approximation to the interface state $u(\tau, \mathbf{x}_{G_\beta})$ at the Gaussian point $(\tau, \mathbf{x}_{G_\beta})$ is computed in the form of a Taylor series in time, where τ is the local time $\tau = t - t^n$. Working in a local coordinate system x_n for \mathbf{x}_{G_β} , the interface state $u(\tau, 0)$ (i.e. the state at $x_n = 0$) is computed by the Taylor series expansion around local time $\tau = 0$, so that

$$u(\tau, 0) \approx u_k(\tau, 0) := u(0, 0) + \sum_{m=1}^{k-1} \frac{\tau^m}{m!} \partial_t^{(m)} u(0, 0), \quad (4.8)$$

where we take k to be the local approximation order of the polyharmonic spline interpolant used in the WENO reconstruction. Equation (4.8) is used in computing u_k at an intermediate time $t_{G_\alpha} \in [t^n, t^{n+1}]$.

We now turn to the computation of each term on the right hand side of (4.8).

The leading term

The leading term $u(0,0)$ in the expansion (4.8) represents the initial interaction of the initial data via the conservation law, and it is the solution of the following Riemann problem [133, 73]

$$u_t + \partial_{\mathbf{n}}F(u) = 0, \quad (4.9)$$

$$u(0, x_{\mathbf{n}}) = \begin{cases} u_L = \lim_{\mathbf{x} \rightarrow \mathbf{x}_{G_\beta}^-} s_L(\mathbf{x}), & x_{\mathbf{n}} < 0, \\ u_R = \lim_{\mathbf{x} \rightarrow \mathbf{x}_{G_\beta}^+} s_R(\mathbf{x}), & x_{\mathbf{n}} > 0, \end{cases} \quad (4.10)$$

where the *boundary extrapolated* values u_L and u_R are obtained by evaluating the polynomial spline interpolant inside and outside the actual cell interface at the Gaussian point \mathbf{x}_{G_β} . We will refer to u_L and u_R as the *left* and *right* states, respectively. The term $u(0,0)$, sometimes denoted u^* , is commonly known as the *Godunov state*. We note that the use of u^* alone in approximation $u(\tau,0)$ will result in a first order upwind finite volume method like the Godunov method.

The higher order derivatives

The next step in the formulation of the ADER-WENO method is to compute the coefficients of the higher order terms in (4.8) in time, i.e. we need to compute $\partial_t^{(m)}u(t, \mathbf{x})$, $m = 1, \dots, k-1$ at the local interface $(0,0)$. The ADER-WENO method relies on replacing all the time derivatives with the spatial derivatives by repeated use of the governing differential equation via a technique known as the Cauchy-Kowalewski procedure, or Lax-Wendroff procedure [76]. Hence,

$$\partial_t^{(m)}u(t, \mathbf{x}) = \mathbf{H}^m(D^{(0)}u, D^{(1)}u, \dots, D^{(m)}u), \quad m = 1, \dots, k-1,$$

where $D^{(j)}u$ is the set of all j th spatial derivatives of u .

To demonstrate the Cauchy-Kowalewski procedure, we need to re-write the governing PDE (4.1) as

$$u_t + f_1'(u)u_{x_1} + f_2'(u)u_{x_2} = 0, \quad (4.11)$$

where $f_i'(u) = \frac{\partial f_i(u)}{\partial u}$, $i = 1, 2$. Thus, the first time derivative is obtained from (4.11) as

$$u_t = -f_1'(u)u_{x_1} - f_2'(u)u_{x_2}. \quad (4.12)$$

We see that the first derivatives in time can be expressed in terms of the first derivatives in space. The higher order time derivatives of u can be expressed as spatial derivatives by successive partial differentiation of (4.12) with respect to t . In this work, we use splines of order at most $k = 3$ and thus require only the first and second order time

derivatives. To this end, we further compute the second derivative

$$u_{tt} = -f_1''(u)u_t u_{x_1} - f_1'(u)u_{tx_1} - f_2''(u)u_t u_{x_2} - f_2'(u)u_{tx_2} \quad (4.13)$$

only. The mixed derivatives u_{tx_1} and u_{tx_2} in (4.13) are obtained by differentiating (4.12) with respect x_1 and x_2 . This yields

$$u_{tx_1} = -f_1''(u)u_{x_1}^2 - f_1'(u)u_{x_1x_1} - f_2''(u)u_{x_1}u_{x_2} - f_2'(u)u_{x_1x_2}, \quad (4.14)$$

$$u_{tx_2} = -f_1''(u)u_{x_1}u_{x_2} - f_1'(u)u_{x_1x_2} - f_2''(u)u_{x_2}^2 - f_2'(u)u_{x_2x_2}. \quad (4.15)$$

We remark that the time derivatives, when converted to spatial derivatives, involve terms like $f'(u)$, $f''(u)$, \dots which could become laborious to compute especially for nonlinear conservation laws. However, using Cauchy-Kowalewski procedure to express the time derivatives in terms of the space derivatives is necessary for the effective implementation and accuracy of the ADER-WENO scheme.

We now need to determine the unknown spatial derivatives u_{x_1} , u_{x_2} , \dots , (instead of the time derivatives) at the quadrature points on the cell interfaces. Before we do this, we first of all note that by successive differentiation of the governing equation (4.11) with respect to x_1 and/or x_2 , we can obtain the evolution equations for each of the spatial derivatives, see [129]. Moreover, the linearized form of (4.11) holds for all the space derivatives $\varphi^\gamma = D^\gamma u$, $0 < |\gamma| \leq k-1$ where $D^\gamma = \frac{\partial^{|\gamma|}}{\partial x_1^{\gamma_1} \partial x_2^{\gamma_2}}$ is the γ -th partial derivative operator. In [130, 73], the Godunov state u^* was used in the linearization of (4.11). Hence, the linearized evolution equations for the derivatives are of the form

$$\varphi_t^\gamma + f_1'(u^*)\varphi_{x_1}^\gamma + f_2'(u^*)\varphi_{x_2}^\gamma = 0, \quad 0 < |\gamma| \leq k-1. \quad (4.16)$$

Equation (4.16) can be re-written as

$$\varphi_t^\gamma + \nabla \cdot F^\gamma(u) = 0, \quad (4.17)$$

where $F^\gamma(u) = (f_1'(u^*)\varphi^\gamma, f_2'(u^*)\varphi^\gamma)^T$.

Following Toro & Titarev [133], the spatial derivatives at \mathbf{x}_{G_β} are obtained by solving a set of linearized conventional Riemann problems. Now, if the *boundary extrapolated* values for the derivatives are given as

$$\begin{aligned} \varphi_L^\gamma &= \lim_{\mathbf{x} \rightarrow \mathbf{x}_{G_\beta}^-} D^\gamma s_L(\mathbf{x}), \\ \varphi_R^\gamma &= \lim_{\mathbf{x} \rightarrow \mathbf{x}_{G_\beta}^+} D^\gamma s_R(\mathbf{x}), \end{aligned} \quad (4.18)$$

then the linearized Riemann problems for the spatial derivatives are of the form

$$\varphi_t^\gamma + \partial_{\mathbf{n}} F^\gamma(u) = 0, \quad (4.19)$$

$$\varphi^\gamma(0, x_{\mathbf{n}}) = \begin{cases} \varphi_L^\gamma, & x_{\mathbf{n}} < 0, \\ \varphi_R^\gamma, & x_{\mathbf{n}} > 0. \end{cases} \quad (4.20)$$

The solution $\varphi^\gamma = D^\gamma u$ of the linear Riemann problem (4.19), (4.20) obviously exists since (4.19) is a linear advection equation with constant coefficients with piecewise smooth initial conditions. Hence all the spatial derivatives at the interface $x_{\mathbf{n}} = 0$ (i.e. at \mathbf{x}_{G_β}) can be computed locally. We use these values to obtain the form of the time derivatives and define the value of $u_k(\tau, \mathbf{x}_{G_\beta})$ as

$$u_k(\tau, \mathbf{x}_{G_\beta}) = c_0 + c_1\tau + c_2\tau^2 + \dots + c_{k-1}\tau^{k-1} \quad (4.21)$$

where the coefficients are

$$c_k = \frac{\partial_t^{(k)} u(0, 0)}{k!}.$$

The numerical flux \tilde{F} in (4.3) is then evaluated as

$$\tilde{F}_{T,j}^n \approx \sum_{\alpha=1}^{q_t} K_\alpha |\partial T_j| \sum_{\beta=1}^{q_x} L_\beta F(u_k(t_{G_\alpha}, \mathbf{x}_{G_\beta})) \cdot \mathbf{n}_j. \quad (4.22)$$

Thus, we solve one Riemann problem, which may be linear or nonlinear depending on the nature of the governing equation, and then solve a sequence of $\frac{1}{2}k(k+1) - 1$ linear Riemann problems at each Gaussian point. It is therefore clear that using the leading term alone will give rise to the Godunov method, while using the higher order terms enables us to compute the flux (4.5) at the desired order of accuracy.

4.1.3 Computing the derivatives

From (4.18) and (4.20), we observe that the derivatives of the recovery function play a key role in the implementation of the ADER-WENO scheme. More specifically, the derivatives evaluated at the cell interface serve as the initial data for a set of GRPs. We stress that care must be taken in the implementation of the derivatives of polyharmonic spline interpolant in a numerically stable fashion as was the case when we implemented the polyharmonic spline reconstruction in Section 3.5.

The need for a strategy for the stable and efficient evaluation of derivative of the polyharmonic spline interpolant motivated the analysis in Subsection 2.3.2. The proposed method of evaluation is an extension of the earlier results of Iske [63, 64] on approximation order and numerical stability of local polyharmonic spline interpolation.

In fact, our numerical experiments reveal that this strategy is particularly important as the mesh becomes finer and when $\phi_{2,3}(r) = r^4 \log(r)$ was used for reconstruction. We believe our results will be relevant in any future implementation of the ADER-WENO method with higher order splines.

In the discussion that follows, as in Section 3.5, we will denote the polyharmonic spline interpolant as s^h and we will use the superscript h on triangles, on the triangulation and on stencils. We will be working in \mathbb{R}^2 for now. Once again, the description of the interpolation problem and the derivatives of the interpolant in this section is the same as in Subsection 2.3.2 but with the notation used in Section 3.5.

Now, suppose we have a triangle $T^h \in \mathcal{T}^h$ with stencil $\mathcal{S}^h = \{T_1^h, T_2^h, \dots, T_n^h\} \subset \mathcal{T}^h$, i.e. $T^h \in \mathcal{S}^h$, then for some fixed point $\mathbf{x}_0 \in \mathbb{R}^2$ and any $h > 0$, we recall that the *scaled* reconstruction problem is

$$\lambda_{T^h} u(\mathbf{x}_0 + h\mathbf{x}) = \lambda_{T^h} s^h(\mathbf{x}_0 + h\mathbf{x}), \quad T^h \in \mathcal{S}^h, \quad (4.23)$$

where the set of cell averages operators $\{\lambda_{T^h}\}_{T^h \in \mathcal{S}^h}$ is \mathcal{P}_k^2 -*unisolvent* and of moderate size. Moreover, s^h denotes the unique polyharmonic spline interpolant of the form

$$s^h(h\mathbf{x}) = \sum_{T^h \in \mathcal{S}^h} c_{T^h}^h \lambda_{T^h}^y \phi_{2,k}(\|h\mathbf{x} - h\mathbf{y}\|) + p(h\mathbf{x}), \quad p \in \mathcal{P}_k^2, \quad (4.24)$$

satisfying (4.23) and where we assume, without loss of generality, that $\mathbf{x}_0 = 0$. The derivative of the polyharmonic spline interpolant is given by

$$D^\gamma s^h(h\mathbf{x}) = \sum_{T^h \in \mathcal{S}^h} c_{T^h}^h \lambda_{T^h}^y D^\gamma \phi_{2,k}(\|h\mathbf{x} - h\mathbf{y}\|) + D^\gamma p(h\mathbf{x}). \quad (4.25)$$

The coefficients $\mathbf{c}^h = \{c_{T^h}\}_{T^h \in \mathcal{S}^h} \in \mathbb{R}^n$, $\mathbf{d}^h = (d_\alpha^h)_{|\alpha| < k} \in \mathbb{R}^q$ of s^h in (4.24) and $D^\gamma s^h$ in (4.25) are obtained by solving

$$\underbrace{\begin{bmatrix} A_h & P_h \\ P_h^T & \mathbf{0} \end{bmatrix}}_{\mathbf{A}_h} \underbrace{\begin{bmatrix} \mathbf{c}^h \\ \mathbf{d}^h \end{bmatrix}}_{\mathbf{b}^h} = \underbrace{\begin{bmatrix} u|_{\mathcal{S}^h} \\ 0 \end{bmatrix}}_{u_h}, \quad (4.26)$$

where

$$A_h = (\lambda_{T^h}^x \lambda_{R^h}^y \phi_{2,k}(\|h\mathbf{x} - h\mathbf{y}\|))_{T^h, R^h \in \mathcal{S}^h} \in \mathbb{R}^{n \times n},$$

$$P_h = (\lambda_{T^h}(h\mathbf{x})^\alpha)_{T^h \in \mathcal{S}^h, 0 \leq |\alpha| < k} \in \mathbb{R}^{n \times q},$$

and $u|_{\mathcal{S}^h} = (\lambda_{T^h}(u))_{T^h \in \mathcal{S}^h} \in \mathbb{R}^n$.

Suppose the Lagrange-type representation of the s^h is given as

$$s^h(h\mathbf{x}) = \sum_{T^h \in \mathcal{S}^h} \ell_{T^h}^h(h\mathbf{x}) \lambda_{T^h} u(h\mathbf{x}) \quad (4.27)$$

with the Lagrange basis functions $\ell_{T^h}^h$ satisfying

$$\lambda_{R^h}(\ell_{T^h}^h(h\mathbf{x})) = \begin{cases} 1, & \text{for } T^h = R^h; \\ 0, & \text{for } T^h \neq R^h, \end{cases} \quad T^h, R^h \in \mathcal{S}^h, \quad (4.28)$$

then

$$D^\gamma s^h(h\mathbf{x}) = \sum_{T^h \in \mathcal{S}^h} D^\gamma \ell_{T^h}^h(h\mathbf{x}) \lambda_{T^h} u(h\mathbf{x}). \quad (4.29)$$

The vector

$$D^\gamma \ell^h(h\mathbf{x}) = (D^\gamma \ell_{T^h}^h(h\mathbf{x}))_{T^h \in \mathcal{S}^h} \in \mathbb{R}^n,$$

together with

$$D^\gamma v^h(h\mathbf{x}) = (D^\gamma v_1^h(h\mathbf{x}), \dots, D^\gamma v_q^h(h\mathbf{x}))^T,$$

are the unique solution of the linear system

$$\underbrace{\begin{bmatrix} A_h & P_h \\ P_h^T & \mathbf{0} \end{bmatrix}}_{\mathbf{A}_h} \underbrace{\begin{bmatrix} D^\gamma \ell^h(h\mathbf{x}) \\ D^\gamma v^h(h\mathbf{x}) \end{bmatrix}}_{D^\gamma v^h(h\mathbf{x})} = \underbrace{\begin{bmatrix} D^\gamma R_h(h\mathbf{x}) \\ D^\gamma S_h(h\mathbf{x}) \end{bmatrix}}_{D^\gamma \beta_h(h\mathbf{x})}, \quad (4.30)$$

where

$$D^\gamma R_h(h\mathbf{x}) = (\lambda_{T^h}^y D^\gamma \phi_{2,k}(\|h\mathbf{x} - h\mathbf{y}\|))_{T^h \in \mathcal{S}^h} \in \mathbb{R}^n$$

$$D^\gamma S_h(h\mathbf{x}) = (D^\gamma (h\mathbf{x})^\alpha)_{|\alpha| < k} \in \mathbb{R}^q.$$

The system (4.30) is obtained by the differentiation of (3.29), see [144].

We recall from Lemma 2.22 that

$$D^\gamma \ell^h(h\mathbf{x}) = h^{-|\gamma|} D^\gamma \ell^1(\mathbf{x}) \quad \text{for every } \mathbf{x} \in \mathbb{R}^2 \quad \text{and} \quad |\gamma| < k,$$

and so in line with Subsection 2.3.2, the stable evaluation of $D^\gamma s^h$ is given as

$$\begin{aligned}
D^\gamma s^h(h\mathbf{x}) &= \langle D^\gamma \ell^h(h\mathbf{x}), u|_{\mathcal{S}^h} \rangle \\
&= \langle h^{-|\gamma|} D^\gamma \ell^1(\mathbf{x}), u|_{\mathcal{S}^h} \rangle \\
&= h^{-|\gamma|} \langle D^\gamma \nu^1(\mathbf{x}), u_h \rangle \\
&= h^{-|\gamma|} \langle \mathbf{A}_1^{-1} D^\gamma \beta_1(\mathbf{x}), u_h \rangle \\
&= h^{-|\gamma|} \langle D^\gamma \beta_1(\mathbf{x}), \mathbf{A}_1^{-1} u_h \rangle.
\end{aligned} \tag{4.31}$$

We will use the final expression in (4.31) for the evaluation of the derivative of the polyharmonic spline interpolant at the cell interface.

4.1.4 Algorithm for the ADER-WENO method

Once an approximation for $u(t_{G_\alpha}, \mathbf{x}_{G_\beta})$ is obtained using (4.21), the flux can then be computed via (4.5) and the cell averages can be updated via the two-level explicit scheme (4.3). In summary, the ADER-WENO algorithm for advancing one step in time from t^n to t^{n+1} is given as follows.

Algorithm 4.1 (The ADER-WENO method)

Input: *Triangulation \mathcal{T} with cell averages \bar{u}_T^n , $T \in \mathcal{T}$, time step $\Delta t > 0$ and the order k .*

- *Compute the polyharmonic splines of order k from cell averages \bar{u}_T^n using the WENO reconstruction in Algorithm 3.1.*
- **FOR** *each $T \in \mathcal{T}$ DO*
 - (1) *Use the Cauchy-Kowalewski procedure to express the time derivatives in terms of the space derivatives.*
 - (2) *Solve the one-dimensional GRPs at the Gaussian integration points \mathbf{x}_{G_β} at the cell interface.*
 - (3) *Obtain the solution u at the time Gaussian integration points t_{G_α} via (4.8).*
 - (4) *Compute the numerical fluxes $\tilde{F}_{T,j}$, $j = 1, 2, 3$ using (4.5).*

(5) Update cell average values \bar{u}_T^{n+1} using (4.3).

Output: Updated cell average values \bar{u}_T^{n+1} for all $T \in \mathcal{T}$.

4.2 Numerical Examples

In this section, we consider the same numerical examples as treated in Section 3.8. To this end, we once again consider the linear advection equation

$$u_t + u_{x_1} + u_{x_2} = 0, \quad \text{for } u \equiv u(t, \mathbf{x}) \quad \text{with } \mathbf{x} = (x_1, x_2) \in \mathbb{R}^2 \quad (4.32)$$

with the initial condition

$$u_0(\mathbf{x}) = u(0, \mathbf{x}) = \sin^2\left(\pi\left(x_1 + \frac{1}{2}\right)\right) \sin^2\left(\pi\left(x_2 + \frac{1}{2}\right)\right) \quad (4.33)$$

on the computational domain $\Omega = [-0.5, 0.5] \times [-0.5, 0.5] \subset \mathbb{R}^2$ and the time interval $I = [0, 1]$. We also consider the Burgers equation

$$u_t + \left(\frac{1}{2}u^2\right)_{x_1} + \left(\frac{1}{2}u^2\right)_{x_2} = 0, \quad (4.34)$$

with the initial condition

$$u_0(\mathbf{x}) = u(0, \mathbf{x}) = \frac{1}{4} + \frac{1}{2} \sin(\pi(x_1 + x_2)) \quad (4.35)$$

on the computational domain $\Omega = [-1, 1] \times [-1, 1] \subset \mathbb{R}^2$. The computations for Burgers' equation are carried out on the time interval $[0, 0.1]$.

We solve both problems using the ADER-WENO method for which the WENO reconstruction is performed with $\phi_{2,2}(r) = r^2 \log(r)$ (thin plate splines) and $\phi_{2,3}(r) = r^4 \log(r)$ only. We use the same stencil sizes as in Chapter 3, i.e we use seven stencils of size four for thin plate spline reconstruction and nine stencils of size nine for reconstruction with $\phi_{2,3}(r) = r^4 \log(r)$. We use the CFL number $c = 0.5$ in all our computations. We record the errors $E_p(h)$ together with the convergence rates k_p for $p = 1, 2, \infty$.

For both the linear advection equation and Burgers' equation we observe second and third order convergence for the ADER-WENO method with reconstruction using $\phi_{2,2}(r) = r^2 \log(r)$ and $\phi_{2,3}(r) = r^4 \log(r)$ respectively as shown in Tables 4.1, 4.2, 4.3 and 4.4. Moreover, the error decreases by an order of magnitude when the formal order of accuracy increases. We therefore advocate higher order implementations of the ADER-WENO method, as they can be a contender for very high order time stepping

h	$E_1(h)$	k_1	$E_2(h)$	k_2	$E_\infty(h)$	k_∞
1/8	$5.3421 \cdot 10^{-2}$	—	$6.9206 \cdot 10^{-2}$	—	$1.9900 \cdot 10^{-1}$	—
1/16	$9.9088 \cdot 10^{-3}$	2.44	$1.3099 \cdot 10^{-2}$	2.40	$4.2914 \cdot 10^{-2}$	2.21
1/32	$2.1862 \cdot 10^{-3}$	2.17	$3.0085 \cdot 10^{-3}$	2.12	$1.1495 \cdot 10^{-2}$	1.90
1/64	$5.3530 \cdot 10^{-4}$	2.03	$7.3156 \cdot 10^{-4}$	2.04	$3.1629 \cdot 10^{-3}$	1.86
1/128	$1.2837 \cdot 10^{-4}$	2.06	$1.7544 \cdot 10^{-4}$	2.07	$8.3581 \cdot 10^{-4}$	1.92

Table 4.1: Linear advection. Results by ADER-WENO method using $\phi_{2,2}(r) = r^2 \log(r)$ reconstruction.

h	$E_1(h)$	k_1	$E_2(h)$	k_2	$E_\infty(h)$	k_∞
1/8	$6.1728 \cdot 10^{-2}$	—	$7.3862 \cdot 10^{-2}$	—	$1.8427 \cdot 10^{-1}$	—
1/16	$7.6627 \cdot 10^{-3}$	3.01	$9.3616 \cdot 10^{-3}$	2.98	$2.5205 \cdot 10^{-2}$	2.87
1/32	$1.0337 \cdot 10^{-3}$	2.89	$1.2984 \cdot 10^{-3}$	2.85	$3.5693 \cdot 10^{-2}$	2.82
1/64	$1.3658 \cdot 10^{-4}$	2.92	$1.6920 \cdot 10^{-4}$	2.94	$5.1608 \cdot 10^{-4}$	2.80
1/128	$1.7192 \cdot 10^{-5}$	2.99	$2.2355 \cdot 10^{-5}$	2.92	$7.0106 \cdot 10^{-5}$	2.88

Table 4.2: Linear advection. Results by ADER-WENO method using $\phi_{2,3}(r) = r^4 \log(r)$ reconstruction.

h	$E_1(h)$	k_1	$E_2(h)$	k_2	$E_\infty(h)$	k_∞
1/8	$8.1771 \cdot 10^{-2}$	—	$6.2890 \cdot 10^{-2}$	—	$1.8409 \cdot 10^{-1}$	—
1/16	$1.8045 \cdot 10^{-2}$	2.18	$1.4170 \cdot 10^{-2}$	2.15	$4.3240 \cdot 10^{-2}$	2.09
1/32	$4.4844 \cdot 10^{-3}$	2.00	$3.4782 \cdot 10^{-3}$	1.92	$1.2590 \cdot 10^{-2}$	1.78
1/64	$1.0862 \cdot 10^{-3}$	2.04	$1.1468 \cdot 10^{-3}$	1.78	$3.4926 \cdot 10^{-3}$	1.85
1/128	$2.6412 \cdot 10^{-4}$	2.04	$2.8472 \cdot 10^{-4}$	2.01	$9.4887 \cdot 10^{-4}$	1.88

Table 4.3: Burgers' equation. Results by ADER-WENO method using $\phi_{2,2}(r) = r^2 \log(r)$ reconstruction.

h	$E_1(h)$	k_1	$E_2(h)$	k_2	$E_\infty(h)$	k_∞
1/8	$6.3527 \cdot 10^{-2}$	—	$4.8526 \cdot 10^{-2}$	—	$1.5243 \cdot 10^{-1}$	—
1/16	$9.9128 \cdot 10^{-3}$	2.68	$7.3650 \cdot 10^{-3}$	2.72	$2.4624 \cdot 10^{-2}$	2.63
1/32	$1.3373 \cdot 10^{-3}$	2.89	$1.0215 \cdot 10^{-3}$	2.85	$3.4390 \cdot 10^{-3}$	2.84
1/64	$1.8292 \cdot 10^{-4}$	2.87	$1.3497 \cdot 10^{-4}$	2.92	$4.6394 \cdot 10^{-4}$	2.89
1/128	$2.3672 \cdot 10^{-5}$	2.95	$1.7106 \cdot 10^{-5}$	2.98	$8.6155 \cdot 10^{-5}$	2.81

Table 4.4: Burgers' equation. Results by ADER-WENO method using $\phi_{2,3}(r) = r^4 \log(r)$ reconstruction.

methods.

4.3 Smolarkiewicz's Deformational Flow Test

We test the robustness of the ADER-WENO method by applying it to a variable-coefficient linear advection equation of the form (3.45) that possesses a multiscale behaviour. The problem we wish to solve here is known as Smolarkiewicz's deformational flow test [113] and is often used for a qualitative evaluation of advection schemes for atmospheric flow simulations. According to Smolarkiewicz [113], the deformational flow test is a convenient tool for studying a solution's accuracy on the resolved scales and for addressing the questions of nonlinear stability due to the existence of unresolved scales. The highly deformational velocity field of this test specifies symmetrical counter-rotating vortices and is given by

$$\sigma_1(\mathbf{x}) = Ak \sin(kx_1) \sin(kx_2) \quad \text{and} \quad \sigma_2(\mathbf{x}) = Ak \cos(kx_1) \cos(kx_2),$$

where $k = 4\pi/L$, $A = 8$ and $L = 100$ units. The computational domain is a square of side $L = 100$ units. The initial condition is a cone of height 1 unit and radius 15 units and it is initialised in the centre of the computational domain. The initial condition can be written as

$$u_0(\mathbf{x}) = \begin{cases} 1 - \frac{r}{15}, & r \leq 15; \\ 0, & \text{otherwise,} \end{cases} \quad (4.36)$$

where $r = \sqrt{(x_1 - 50)^2 + (x_2 - 50)^2}$. The radius of the base of the cone is slightly greater than the radius of the vortices, so at initial time, the cone belongs to the area of six vortices, but its main part belongs to the area of the two central ones. Figure 4.3(a)

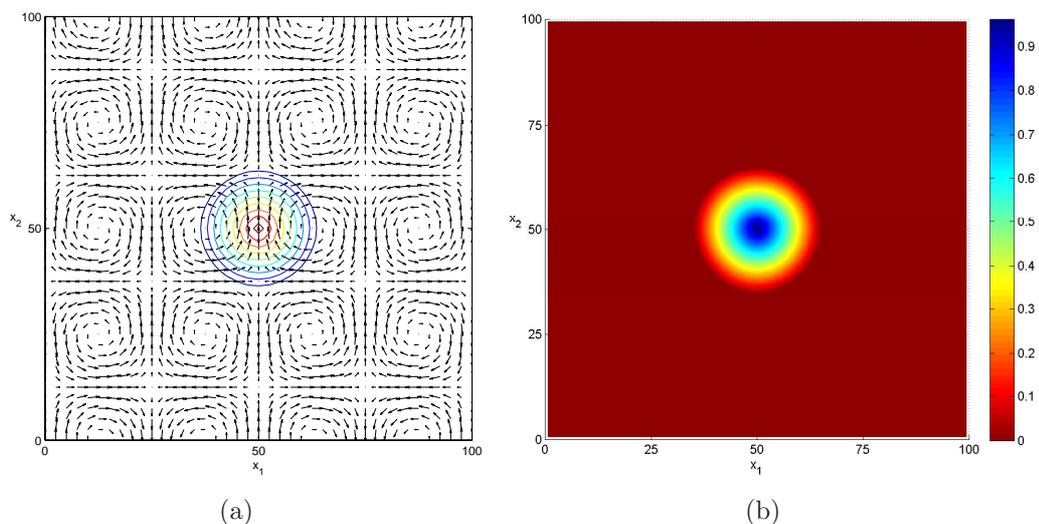


Figure 4.3: (a) Velocity field and (b) initial condition for Smolarkiewicz's deformational test.

shows the velocity field and the contour plot for the initial condition while Figure 4.3(b)

is a plot of the initial profile.

The analytical solution which was provided by Staniforth *et al* [120] consists of spiral distributions which wind tighter and tighter within the square vortex cells. As the solution evolves in time, filaments are produced within the distribution which may become too thin to be resolved by the grid.

We will use the ADER-WENO method with thin plate spline reconstruction to solve the Smolarkiewicz's deformational flow test on an unstructured triangular grid. We use a mesh of size $h = 1$ and time step $\Delta t = 0.26376$. Figure 4.4 shows the solution at time $3T/200$, $3T/100$, $9T/200$, $3T/50$, $4T/50$ and $T/10$ ($T = 2637.6$ is the final time used in [113]).

Staniforth *et al* [120] pointed out that for a mesh of resolution 1, used in [113] and here, the numerical solution is valid only for time $t \leq T/50$. After time $t = T/50$, the features of the solution become too fine to be captured by a mesh of width $h = 1$. In Figure 4.5, we display cross sections of the computed solution and the exact solution along the middle of the computational domain. We show this comparison with the exact solution for times $t = T/100$ and $t = 3T/200$ (before $t = T/50$) and for time $3T/50$ and $T/10$ (after $t = T/50$). We notice that the approximate solution is in good agreement with the exact solution before time $t = T/50$. Some numerical diffusion is observed at time $3T/50$ and the numerical solution becomes more diffusive at time $T/10$.

In conclusion, this numerical experiment shows that the ADER-WENO method provides satisfactory results in the resolution of discontinuities. It is also stable and only a limited amount of diffusion witnessed even when the simulation had advanced in time and the solution became increasingly multiscaled.

4.4 Some Comments on the RK-WENO and ADER-WENO Schemes

The main difference between the RK-WENO method described in Chapter 3 and the ADER-WENO method lies in the time-stepping and flux evaluation procedures. The RK-WENO method uses an SSP Runge-Kutta time stepping in combination with the polyharmonic spline WENO reconstruction technique and the Lax-Friedrich flux while the ADER-WENO scheme combines the WENO reconstruction algorithm with the ADER time discretisation and flux evaluation procedure.

The use of a multi-stage Runge-Kutta method in the RK-WENO method means that we need to do a reconstruction for each of the s stages of the Runge-Kutta method. In the case of the ADER-WENO method, we do not need these multiple reconstructions: only one reconstruction is needed for each time step. This may seem to be an advan-

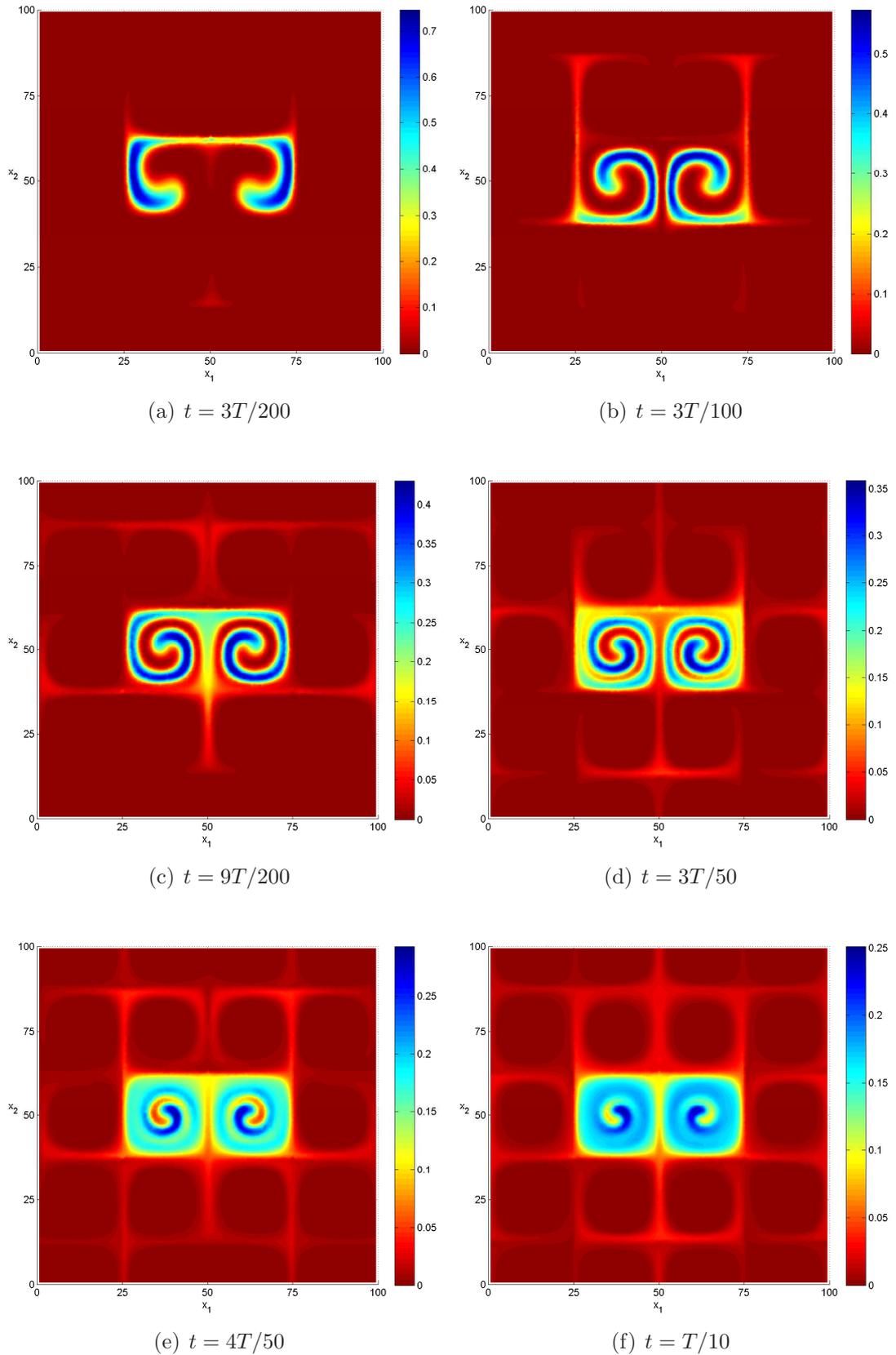
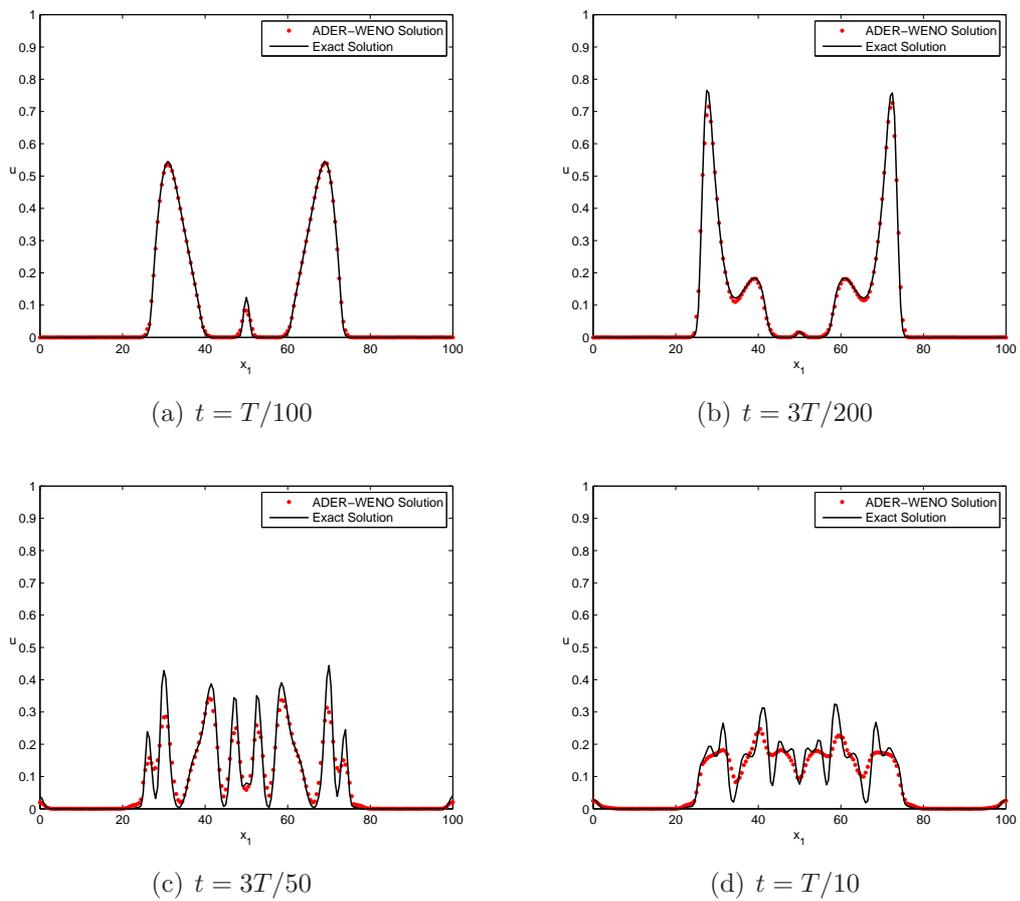


Figure 4.4: Solution of Smolarkiewicz's deformational test using ADER-WENO method at six different times (a) $t = 3T/200$, (b) $t = 3T/100$, (c) $t = 9T/200$, (d) $t = 3T/50$, (e) $t = 4T/50$ and (f) $t = T/10$ where $T = 2637.6$.

Figure 4.5: Cuts along the x_2 -axis for Smolarkiewicz's deformational flow.

tage in terms of computational cost for the ADER-WENO method, but our numerical experiments reveal that a lot of time is spent on computing the derivatives of the polyharmonic spline interpolant. To be precise, for a polyharmonic spline of order k , we need to compute $N_d = \frac{1}{2}k(k+1) - 1$ derivatives. The time used in computing these derivatives is comparable to the time used for each reconstruction step and k increases $N_d \gg s$. Moreover, each of these derivatives must also be carefully processed to ensure that they do not affect the stability and accuracy of the ADER-WENO scheme. For the RK-WENO method, we use the same preconditioning technique at all stages.

We still advocate further work on the ADER-WENO scheme because of the possibility of extending them to arbitrary high order in space and time by adding higher order terms to (4.8) so that the order may be limited only by the available computing resources.

Chapter 5

Adaptivity

Over the past few decades, the use of adaptive methods has become an integral part of many solvers for PDEs. These methods are particularly useful because the PDEs that model a wide variety of phenomena develop multiscale, dynamically singular or nearly singular solutions in localized regions. In particular, solutions of conservation laws may display localized structures like shock waves, contact discontinuities and rarefaction waves or develop steep gradients. Indeed, to enhance the quality of the numerical approximation and reduce the computational costs, numerical methods may require the use of fine resolution over only some portions of the computational domain. Effective and robust adaptive algorithms are particularly useful in multidimensional problems, where the complexity increases exponentially with the dimension.

Finite volume methods are often enhanced with local mesh refinement methods to achieve highly accurate solutions of conservation laws, e.g. [26, 123]. In [12, 99], RBF methods were successfully combined with adaptive strategies for the solution of time dependent partial differential equations. In addition, Li & Hyman [80] combined adaptive mesh refinement with the finite difference WENO method, Smit *et al* [112] implemented grid adaptation with RK-WENO schemes for one dimensional convection dominated PDEs while an adaptive ADER-WENO scheme for scalar conservation laws was provided in [73].

In this chapter, we implement adaptive algorithms using the RK-WENO method of Chapters 3 and the ADER-WENO method of Chapter 4. We first implement a simple stencil adaptivity algorithm which relies on the flexibility in the choice of stencil sizes for polyharmonic spline reconstruction. Next, we will implement mesh adaptation wherein we adaptively modify the computational mesh during the simulation on the basis of an error indicator. This enables us to optimize computing time as well as the use of storage.

In Chapter 6, we will implement an adaptive algorithm based on both mesh refinement and reconstruction order variation.

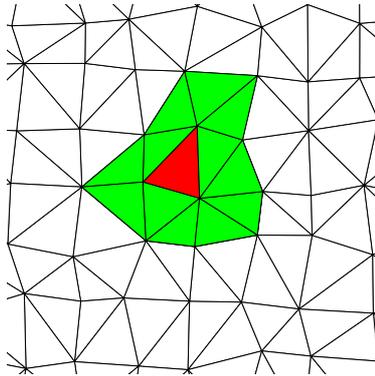


Figure 5.1: A triangle T (marked red) with its Moore neighbours (marked green).

5.1 The Error Indicator

The design and implementation of any adaptive method is usually guided by a suitable error indicator. An error indicator is normally computed for each cell $T \in \mathcal{T}$ and used to detect if a cell lies in a region where the approximation error is large. For elliptic and parabolic problems, there is a well established theory for error estimation, e.g. [8, 90], and there are also some theoretical results available for linear systems of conservation laws [84]. However, there are no standard error estimation results for nonlinear conservation laws. Thus, in the literature, regions where the solution possesses shocks or steep gradients are determined by error indicators which may be more or less heuristic. Tadmor *et al* [89] provided an *a posteriori* error estimate using residuals and the *Lip'* convergence theory. However, their results apply only to one dimensional conservation laws. Löhner *et al* [56] proposed an error indicator based on limiter functions which are computed from higher-order differences. Their indicators are suitable for high resolution schemes that are based on slope limiting because there is usually no additional computational cost. In [118], Sonar & Süli proposed a dual-graph norm error indicator for finite volume approximations of the Euler equations, while Kurganov *et al* [70, 75] used the weak local truncation error in conjunction with quadratic B-splines to develop a smoothness indicator for identifying 'rough' solution regions. Hubbard & Nikiforakis [60] used monitor functions to detect regions of the solution with rich structures like shocks.

In this work, polyharmonic spline interpolation will be used in computing an error indicator for each triangle of a triangulation \mathcal{T} . This is an extension of the indicator used in [48, 73]. In order to compute this error indicator, we first of all assume that each cell average value \bar{u}_T , $T \in \mathcal{T}$ is assigned to the barycenter \mathbf{b}_T of the cell T , i.e. $\bar{u}_T \equiv \bar{u}(\mathbf{b}_T)$. We then compute a polyharmonic spline interpolant s of the form

$$s(\mathbf{x}) = \sum_{T' \in \mathcal{N}_M(T)} c_{T'} \phi_{2,k}(\|\mathbf{x} - \mathbf{b}_{T'}\|) + p(\mathbf{x}), \quad p \in \mathcal{P}_k^2, \quad (5.1)$$

where the barycenters $\mathbf{b}_{T'}$ of the triangles in the Moore neighbourhood $\mathcal{N}_M(T)$ of T (see Figure 5.1 where T is marked red) are regarded as the interpolation points, i.e. s satisfies the interpolation condition

$$s(\mathbf{b}_{T'}) = \bar{u}(\mathbf{b}_{T'}) \quad \text{for all } T' \in \mathcal{N}_M(T).$$

Note that the Moore neighbourhood does not include T itself so that $\bar{u}(\mathbf{b}_{T'}) \neq s(\mathbf{b}_{T'})$ in general. The error indicator is then defined as

$$\varepsilon_T = |\bar{u}(\mathbf{b}_T) - s(\mathbf{b}_T)|. \quad (5.2)$$

Therefore, the error indicator $\varepsilon : \mathcal{T} \mapsto \mathbb{R}$ estimates the local approximation behaviour in the neighbourhood of each triangle in $\mathcal{T}(t)$. A large value of ε_T indicates a large approximation error around T , while a small value of ε_T indicates a small approximation error around T . In our numerical examples, this error indicator was able to effectively locate discontinuities and sharp gradients in the solution u .

5.2 Stencil Adaptivity

In Subsection 3.6.1, we highlighted the flexibility of the stencil selection algorithm in the polyharmonic spline reconstruction method. In fact, we only have a lower bound for the stencil size so as to obtain non-trivial recovery. There is actually no upper bound for the stencil size, although for the sake of computational cost we generally try to keep it relatively small. We can use this flexibility in stencil selection to adaptively adjust the size of the stencil used for recovery on different cells during the simulation. When using thin plate splines for the WENO reconstruction, we observed that stencils of size four were suitable for smooth regions of the solution while stencils of size seven were preferred in regions where the solution has strong variation. To this end, we propose a simple stencil adaptivity strategy on the basis of the size of the error indicator ε_T as follows.

Definition 5.1 *Suppose $\varepsilon^* = \max_{T \in \mathcal{T}} \varepsilon_T$, and let ϑ be a threshold value satisfying $0 < \vartheta < 1$. On cells with $\varepsilon_T > \vartheta \cdot \varepsilon^*$, we use stencils of size seven for the thin plate spline WENO reconstruction otherwise we use stencils of size four.*

In our numerical experiments, we use $\vartheta = 0.05$.

We will use the stencil adaptivity in the WENO reconstruction of the RK-WENO method to solve the linear advection equation and Burgers' equation and compare the results with those obtained when fixed stencil sizes were used.

5.2.1 Example : Linear advection

We solve the linear advection equation (3.35) with initial data [39]

$$u_0(\mathbf{x}) = \begin{cases} \exp\left(\frac{\|\mathbf{x}-\mathbf{c}\|^2}{\|\mathbf{x}-\mathbf{c}\|^2-R^2}\right), & \|\mathbf{x}-\mathbf{c}\| < R; \\ 0, & \text{otherwise,} \end{cases} \quad (5.3)$$

with $R = 0.15$, $\mathbf{c} = (-0.2, -0.2)^T$ on the computational domain $\Omega = [-0.5, 0.5] \times [-0.5, 0.5] \subset \mathbb{R}^2$ using the RK-WENO method. The WENO reconstruction is done with both stencil adaptivity and fixed stencil sizes and we show the errors in the L_1 -, L_2 - and L_∞ -norms at time $t = 0.25$ on meshes of sizes $h = \frac{1}{16}$, $h = \frac{1}{32}$ and $h = \frac{1}{64}$ with 512, 2048 and 8192 triangles respectively.

(a)

Fixed stencil sizes					
h	E_1	E_2	E_∞	N_4	N_7
1/16	$2.0122 \cdot 10^{-2}$	$6.1310 \cdot 10^{-2}$	$3.6325 \cdot 10^{-1}$	—	512
1/32	$7.9146 \cdot 10^{-3}$	$2.6562 \cdot 10^{-2}$	$1.9183 \cdot 10^{-1}$	—	2048
1/64	$2.3427 \cdot 10^{-3}$	$1.0529 \cdot 10^{-2}$	$8.0493 \cdot 10^{-2}$	—	8192

(b)

Adapted stencils					
h	E_1	E_2	E_∞	N_4	N_7
1/16	$1.9748 \cdot 10^{-2}$	$6.0187 \cdot 10^{-2}$	$3.6077 \cdot 10^{-1}$	433	79
1/32	$7.4548 \cdot 10^{-3}$	$2.5282 \cdot 10^{-2}$	$1.5023 \cdot 10^{-1}$	1858	190
1/64	$1.8588 \cdot 10^{-3}$	$8.2181 \cdot 10^{-3}$	$8.0328 \cdot 10^{-2}$	7456	736

Table 5.1: Comparing the solution of the linear advection equation on (a) fixed stencil sizes and on (b) adapted stencils.

We see from Table 5.1, which shows the results at $t = 0.25$, that there is little difference in the errors when we use fixed stencil sizes and when we use variable stencil sizes. In fact, we observed a slight improvement when stencil adaptivity was used. However, we significantly reduce the computational time by using smaller stencils on a large number of cells. Note that in Table 5.1 and the rest of the tables in this chapter, N_4 denotes the number of cells where stencils of size four were used for reconstruction while N_7 denotes the number of cells where stencils of size seven were used. E_1 , E_2 and E_∞ denote the errors in the L_1 -, L_2 - and L_∞ -norms respectively.

5.2.2 Example : Burgers' equation

We further demonstrate the benefits of stencil adaptivity by solving Burgers' equation (3.37) with initial condition (5.3) on the computational domain $\Omega = [-0.5, 0.5] \times [-0.5, 0.5] \subset \mathbb{R}^2$. We solve this problem using stencil adaptation on fixed meshes with mesh widths $h = \frac{1}{16}$, $h = \frac{1}{32}$ and $h = \frac{1}{64}$ and also using fixed stencil sizes. The simulation is run until time $t = 1.2$.

(a)

Fixed stencil sizes					
h	E_1	E_2	E_∞	N_4	N_7
1/16	$1.2415 \cdot 10^{-2}$	$4.4681 \cdot 10^{-2}$	$4.2769 \cdot 10^{-1}$	—	512
1/32	$7.1652 \cdot 10^{-3}$	$3.3673 \cdot 10^{-2}$	$2.1775 \cdot 10^{-1}$	—	2048
1/64	$2.2880 \cdot 10^{-3}$	$1.6900 \cdot 10^{-2}$	$8.2838 \cdot 10^{-2}$	—	8192

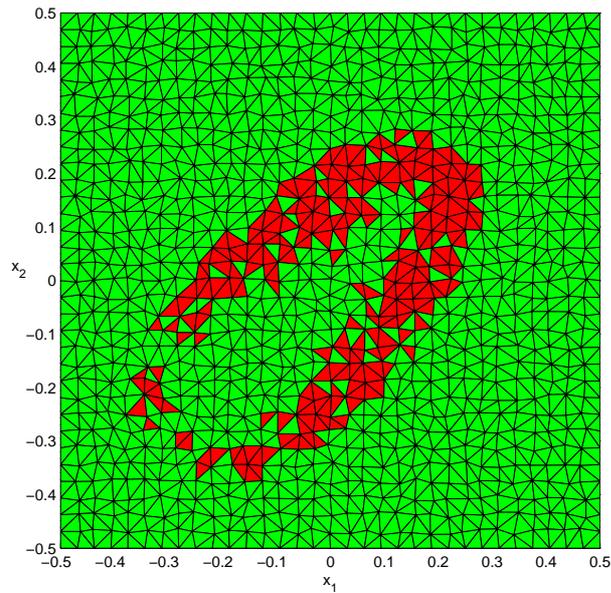
(b)

Adapted stencils					
h	E_1	E_2	E_∞	N_4	N_7
1/16	$1.2318 \cdot 10^{-2}$	$4.4558 \cdot 10^{-2}$	$4.2726 \cdot 10^{-1}$	445	67
1/32	$7.0508 \cdot 10^{-3}$	$3.3647 \cdot 10^{-2}$	$2.1631 \cdot 10^{-1}$	1940	108
1/64	$1.8733 \cdot 10^{-3}$	$1.4550 \cdot 10^{-2}$	$8.2533 \cdot 10^{-2}$	7625	567

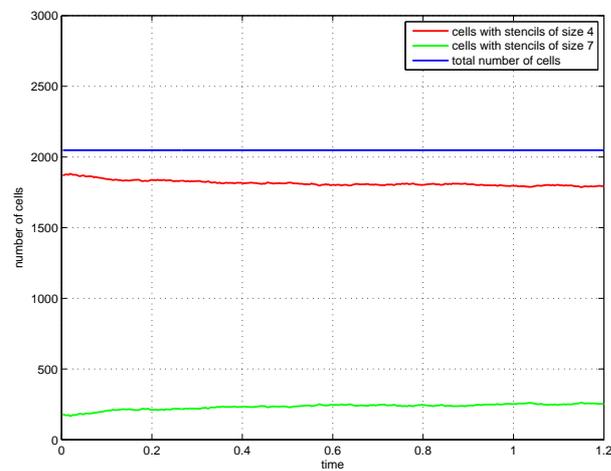
Table 5.2: Comparing the solution of Burgers' equation on (a) fixed stencil sizes and on (b) adapted stencils.

We once again see from Table 5.2 that there is little difference in the errors when fixed stencil sizes are used and when variable stencil sizes are used. Thus, we see that for both the linear advection equation and Burgers' equation, stencil adaptivity does not affect the accuracy of the solution.

Figure 5.2(a) shows the distribution of the stencil sizes for the various cells at time $t = 1.2$ on the mesh of width $h = \frac{1}{32}$. The red cells are those where stencils of size seven were used for the WENO reconstruction, while the green ones are those where stencils of size four were used for the reconstruction. From Figure 5.2(b), which shows the distribution of the stencil sizes throughout the simulation, we notice a slight but steady increase in the number of cells where the reconstruction was done with stencils of size seven as the simulation advances in time. This is because of the increasing support of the solution, which is due to the nonlinearity of the Burgers' equation. This means that there is also a growth in the length of the shock front. In any case, throughout the simulation, the reconstruction of over 80% of the cells was done with stencils of size four, and this can significantly reduce the simulation time.



(a)



(b)

Figure 5.2: (a) Stencil size distribution and (b) number of cells with particular stencil size.

5.3 Mesh Adaptivity

Using an appropriate criterion, the error indicator (5.2) enables us to effectively decide which portions of the computational mesh to refine. Moreover, any practical adaptive mesh method must also concern itself with the derefinement or coarsening of certain regions. This is particularly important for time dependent hyperbolic problems where the shocks or steep gradients necessitating refinement of cells can travel through the whole computational domain leaving a large number of refined elements in their trail as the simulation advances in time which in turn may slow down the computation greatly. Thus, we also need a criterion to decide which cells to coarsen.

The strategy we use in marking cells for refining or coarsening is summarized in the definition below.

Definition 5.2 *Let $\varepsilon^* = \max_{T \in \mathcal{T}} \varepsilon_T$, and let ϑ_r, ϑ_d be two threshold values satisfying $0 < \vartheta_d < \vartheta_r < 1$. We say that a cell $T \in \mathcal{T}$ is to be refined if and only if $\varepsilon_T > \vartheta_r \cdot \varepsilon^*$, and T is coarsened or derefined if and only if $\varepsilon_T < \vartheta_d \cdot \varepsilon^*$.*

In our numerical experiments, we use $\vartheta_r = 0.05$ and $\vartheta_d = 0.01$.

Other methods used in the literature for marking of cells in a triangulation \mathcal{T} for refining or coarsening include the *fixed fraction strategy* e.g. [40] and the *bulking strategy* e.g. [22].

Following [73], a triangular cell $T \in \mathcal{T}$ is refined by inserting its barycenter \mathbf{b}_T as a new node of the triangulation \mathcal{T} . A cell $T \in \mathcal{T}$ is derefined by removing its nodes from the triangulation \mathcal{T} . This means that all cells sharing a node have to be marked for derefinement for the node to be successfully removed from the triangulation. At each time step, after all the new nodes have been inserted and the nodes of the triangles to be coarsened have been removed, the triangulation \mathcal{T} is then updated by a local Delaunay re-triangulation. This enables an adaptive modification of the current triangulation $\mathcal{T}(t)$ yielding a modified triangulation $\mathcal{T}(t + \Delta t)$ at the next time step.

In contrast to computations on a fixed mesh, we have a new mesh at each time step and so we need to compute a new set of Gaussian points. The WENO reconstruction is then performed only after the mesh adaptation has been performed. Therefore, when using the RK-WENO method, if the q Gaussian points for the edge Γ_j of the triangle $T \in \mathcal{T}(t^n)$ are G_1^n, \dots, G_q^n , then $L_T(\bar{u}_T(t))$ in (3.15) is defined as

$$L_T(\bar{u}_T(t)) = -\frac{1}{|T|} \sum_{j=1}^3 |\Gamma_j| \sum_{\ell=1}^q w_\ell \mathcal{F}(u_{\text{in}}(t, G_\ell^n), u_{\text{out}}(t, G_\ell^n)) \cdot \mathbf{n}_j, \quad (5.4)$$

where u_{in} is the recovery function on T and u_{out} is the recovery function on the adjacent cell after the mesh has been modified. The reconstruction and the computation of

L_T for all the stages of the Runge-Kutta method are performed on the modified triangulation to advance the solution from t^n to t^{n+1} . Similarly, in the ADER-WENO method, the reconstruction step and the implementation of the GRP are performed on the modified triangulation. We use the CFL number $c = 1$ in all our computations on mesh adaptivity.

5.3.1 Initial mesh adaptation

Initial mesh adaptation modifies the base mesh so that it conforms to the structure of the initial conditions. It uses the same error indicator and refinement/derefinement strategy that is used during the rest of the simulation. This ensures that the rest of the computation is based on a valid and representative mesh. The major difference, though, is that upon refinement of a mesh, the finer grid is initialized again with newly sampled initial conditions, thus representing it with higher accuracy. The initial mesh refinement procedure is usually applied to the new mesh until either the maximum number of cycles N is reached or until all the error indicators are smaller than some threshold value. In all our computations, we perform the initial mesh refinement till $N = 5$.

We illustrate this with the initial data for the swirling deformation flow problem which is the subject of Subsection 5.4.2. The initial condition is a three-body structure on the computational domain $[0, 1]^2$ and it consists of a smooth hump, a cone and a slotted cylinder.

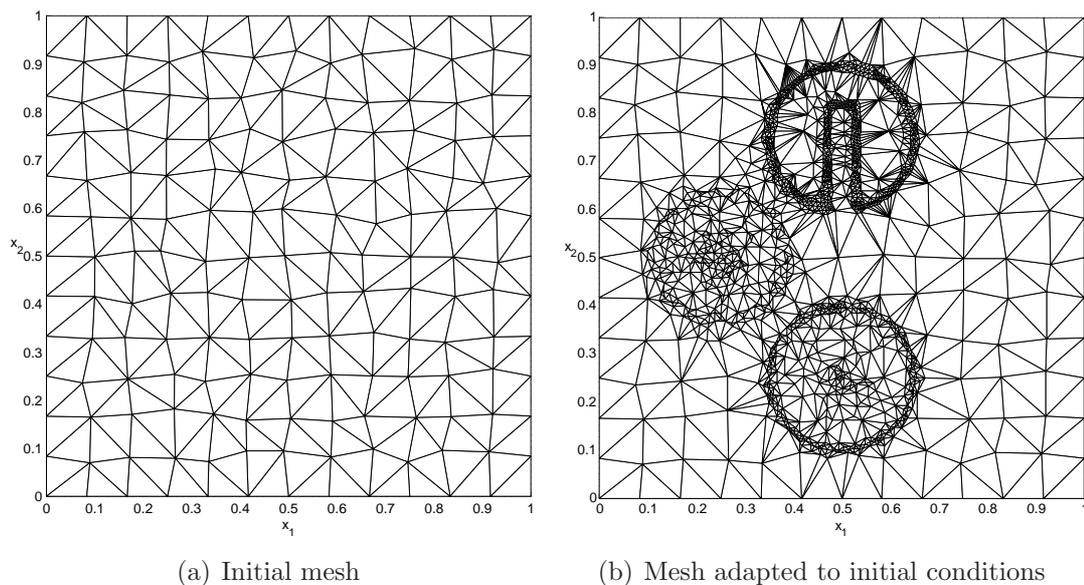


Figure 5.3: (a) A coarse base mesh and (b) the same mesh adapted to the initial condition of the swirling deformation flow problem with 5 refinement cycles.

5.3.2 Conservative interpolation

On any adapted grid, fine meshes appear in certain regions of the domain and so the solution in the fine cells of a grid must be determined from the previous coarser level through an appropriate interpolation method. In addition, mesh derefinement means that some fine grid cells are eliminated and must be averaged in some way to determine the cell averages of the underlying coarser cells. In short, we need to successfully transfer the cell average data on a triangulation $\mathcal{T}(t^n)$ onto a modified output triangulation $\mathcal{T}(t^{n+1})$. It is therefore essential that any interpolation and averaging strategies used do not destroy the essential features of the underlying finite volume method. In particular, our methods are conservative and so we must require that any interpolation and/or averaging maintain numerical conservation whenever possible. In addition, it is necessary to ensure that interpolation does not generate oscillations near discontinuities.

To transfer the cell average data from $\mathcal{T}(t^n)$ to $\mathcal{T}(t^{n+1})$, we use a strategy similar to the one used in [65]. To this end, we first need to compute the intersection of a triangle $T \in \mathcal{T}(t^{n+1})$ and the triangles that it overlaps in $\mathcal{T}(t^n)$. The area which $T \in \mathcal{T}(t^{n+1})$ overlaps a triangle in $\mathcal{T}(t^n)$ is known as a *tile*. These intersection tiles are computed by the chasing algorithm of O'Rourke [91]. We also need to compute the area of the intersection tiles and the mass contained in the tiles.

The mass of a function in a cell T at time t is given as

$$m_T = \int_T u_h(t, \mathbf{x}) d\mathbf{x} \quad (5.5)$$

where u_h is an approximation to the solution of the governing conservation law.

Now, suppose $T \in \mathcal{T}(t^{n+1})$ overlaps l cells in $\mathcal{T}(t^n)$ which we call T_1, \dots, T_l , then

$$T = \bigcup_{T_i \in \mathcal{T}(t^n), i=1, \dots, l} T \cap T_i,$$

where $T \cap T_i$ are the intersection tiles. The mass on each tile is then computed as

$$m_{T \cap T_i} = \int_{T \cap T_i} u_h(t, \mathbf{x}) d\mathbf{x}, \quad i = 1, \dots, l \quad (5.6)$$

where u_h is the restriction of approximate solution at time t^n on $T \cap T_i$. We integrate a non-oscillatory interpolant on $T_i \in \mathcal{T}(t^n)$ over $T \cap T_i$ to compute $m_{T \cap T_i}$. The total mass of $T \in \mathcal{T}(t^{n+1})$ is then given as

$$m_T = \sum_{T_i \in \mathcal{T}(t^n), i=1, \dots, l} m_{T \cap T_i}. \quad (5.7)$$

We now define the cell average of $T \in \mathcal{T}(t^{n+1})$ as

$$\bar{u}_T = \frac{1}{|T|} m_T, \quad \text{for all } T \in \mathcal{T}(t^{n+1}). \quad (5.8)$$

5.3.3 Algorithm

We now provide an algorithm for advancing one step in time from t^n to t^{n+1} using either the adaptive RK-WENO method or the adaptive ADER-WENO method.

Algorithm 5.1 (The adaptive methods)

Input: *Triangulation \mathcal{T} with cell averages \bar{u}_T^n , $T \in \mathcal{T}(t^n)$, threshold values ϑ_r and ϑ_d , the order k of the polyharmonic spline and the method of choice, i.e. RK-WENO or ADER-WENO.*

- *Compute the error indicators for each cell $T \in \mathcal{T}(t^n)$.*
- *Mark cells to be refined or coarsened and perform adaptive modification of triangulation $\mathcal{T}(t^n)$ to produce $\mathcal{T}(t^{n+1})$ using Definition 5.2.*
- *Compute the polyharmonic spline interpolant of order k from cell averages using the WENO reconstruction in Algorithm 3.1.*

FOR each $T \in \mathcal{T}$ **DO**

- **IF** method == RK-WENO **DO**

- (1) *Compute the numerical flux using (3.14).*
- (2) *Form the numerical scheme (3.15).*
- (3) *Update cell average values \bar{u}_T^{n+1} using the SSP Runge-Kutta method (3.17) or (3.18).*

- **ELSEIF** method == ADER-WENO **DO**

- (1) *Use the Cauchy-Kowalewski procedure to express the time derivatives in terms of the space derivatives.*

- (2) Solve the one-dimensional GRPs at the Gaussian integration points \mathbf{x}_{G_β} at the cell interface.
- (3) Obtain the approximation to u at the time Gaussian integration points t_{G_α} via (4.8).
- (4) Compute the numerical fluxes $\tilde{F}_{T,j}$, $j = 1, 2, 3$ using (4.5).
- (5) Update cell average values \bar{u}_T^{n+1} using (4.3).

Output: Updated cell average values \bar{u}_T^{n+1} and a new triangulation $\mathcal{T}(t^{n+1})$.

5.4 Numerical Examples on Mesh Adaptivity

5.4.1 Burgers' equation

We now look at the adaptive solution of the Burgers' equation (3.37) with initial condition (5.3) which is a standard test problem. We note that even for smooth initial data, the solution of Burgers' equation typically develops discontinuities in finite time. We solve this problem with the adaptive RK-WENO method where the reconstruction is performed with $\phi_1(r) = r$, $\phi_{2,2}(r) = r^2 \log(r)$ (thin plate splines) and $\phi_{2,3}(r) = r^4 \log(r)$. We also obtain the numerical solution using the adaptive ADER-WENO scheme for the purpose of comparing with the results from the adaptive RK-WENO method.

In all cases, we start our simulation on a base mesh of 288 triangles and use the strategy in Subsection 5.3.1 to adapt it to the initial condition. The plots for the numerical solution obtained using the RK-WENO method and the ADER-WENO method with the thin plate spline reconstruction, are shown in Figures 5.4 and 5.6 respectively. They are displayed for four different times: $t = 0$, $t = 0.4$, $t = 0.8$ and $t = 1.2$. The corresponding adapted meshes on which the numerical solution by the RK-WENO and ADER-WENO methods were computed are shown in Figures 5.5 and 5.7 respectively.

We notice that the initial condition, which is a Gaussian-shaped function deforms, as the simulation advances in time because of the nonlinearity of the Burgers' equation. The shock is propagated throughout the simulation along the diagonal of the computational domain, and by $t = 0.8$ a very strong shock is present. The shock is however well resolved by the adaptive mesh and in regions where the solution becomes smooth, the mesh is de-refined. This confirms the effectiveness of our adaptive strategy. Figure 5.8(a) shows that for $\phi_{2,2}(r) = r^2 \log(r)$ and $\phi_{2,3}(r) = r^4 \log(r)$, there is a steady increase in the number

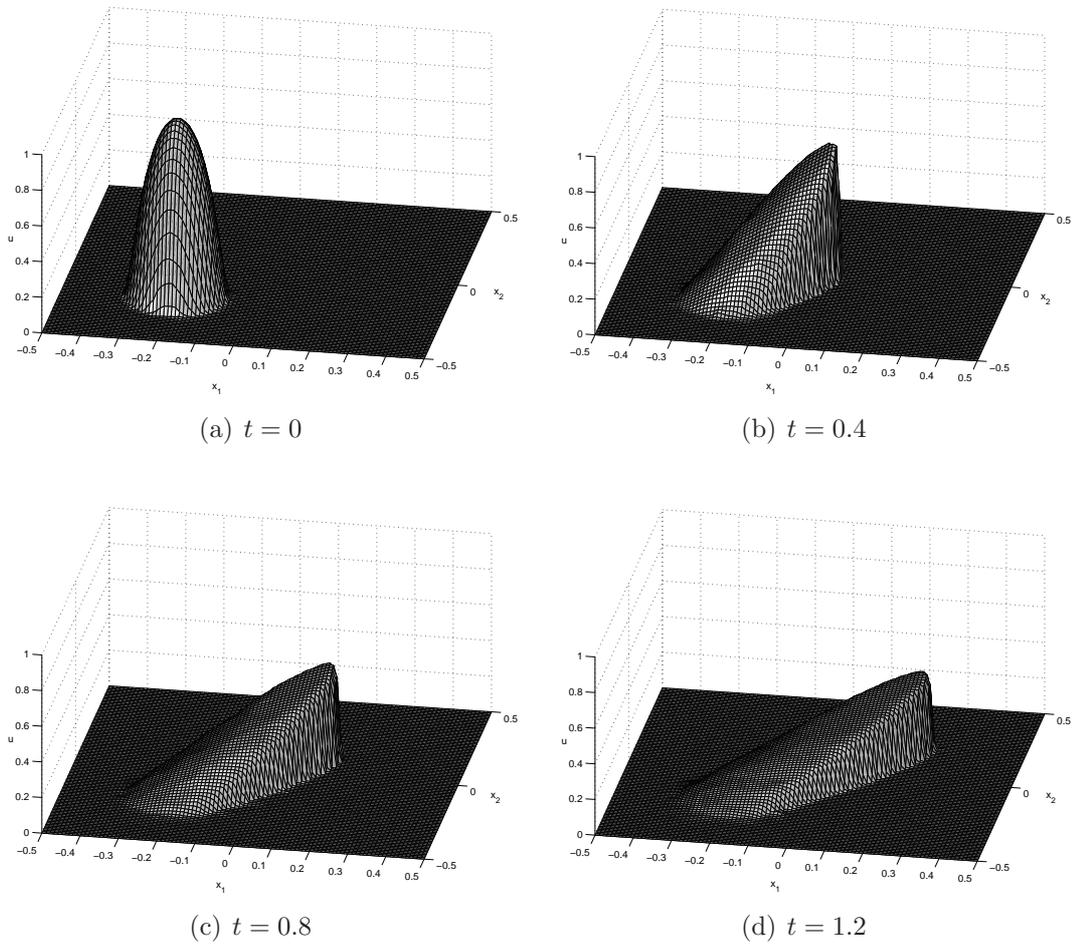


Figure 5.4: Solution of Burgers' equation at times (a) $t = 0$, (b) $t = 0.4$, (c) $t = 0.8$, and (d) $t = 1.2$ using the RK-WENO method.

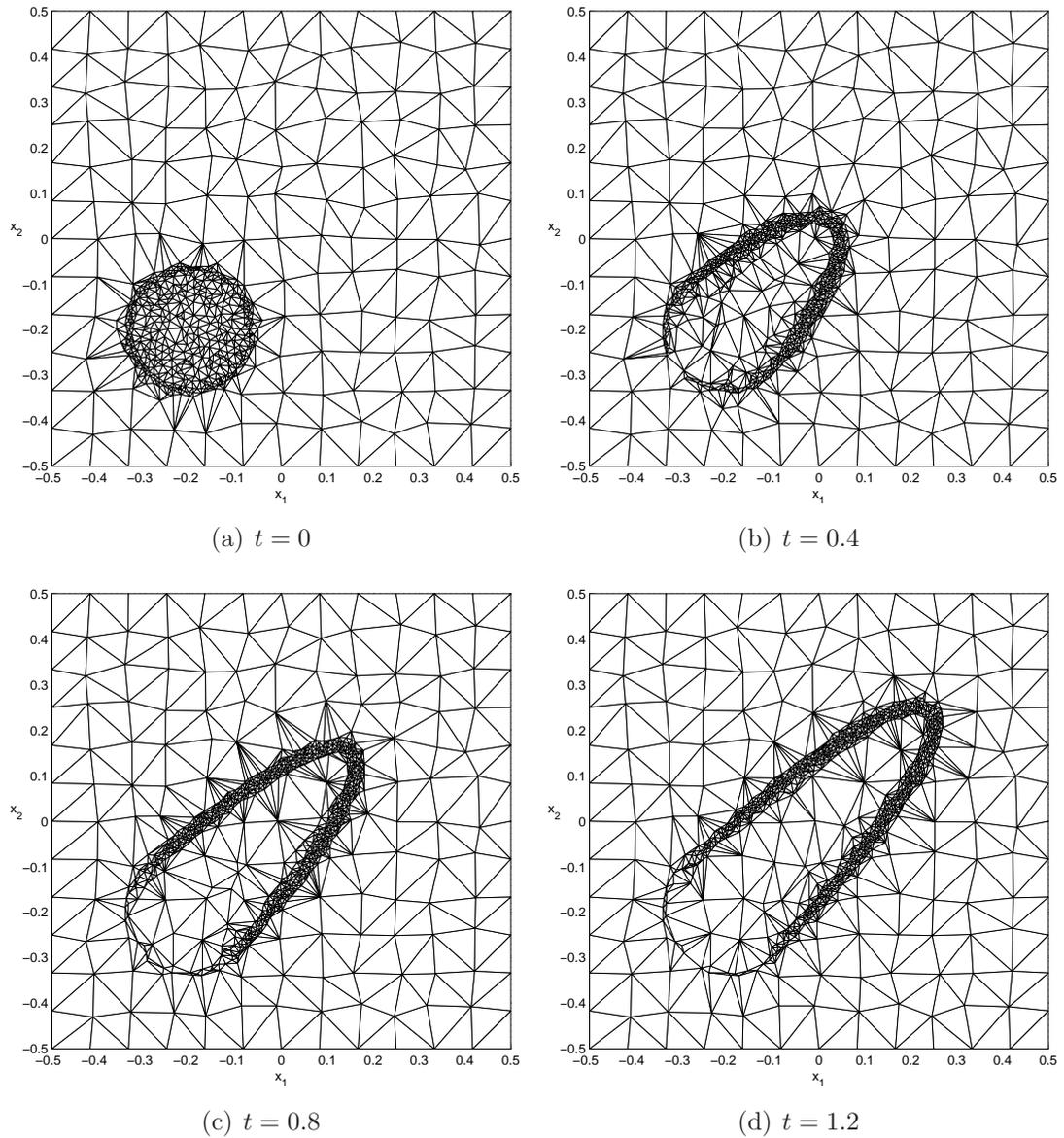


Figure 5.5: Adapted mesh for the solution of Burgers' equation at times (a) $t = 0$, (b) $t = 0.4$, (c) $t = 0.8$, and (d) $t = 1.2$ using the RK-WENO method.

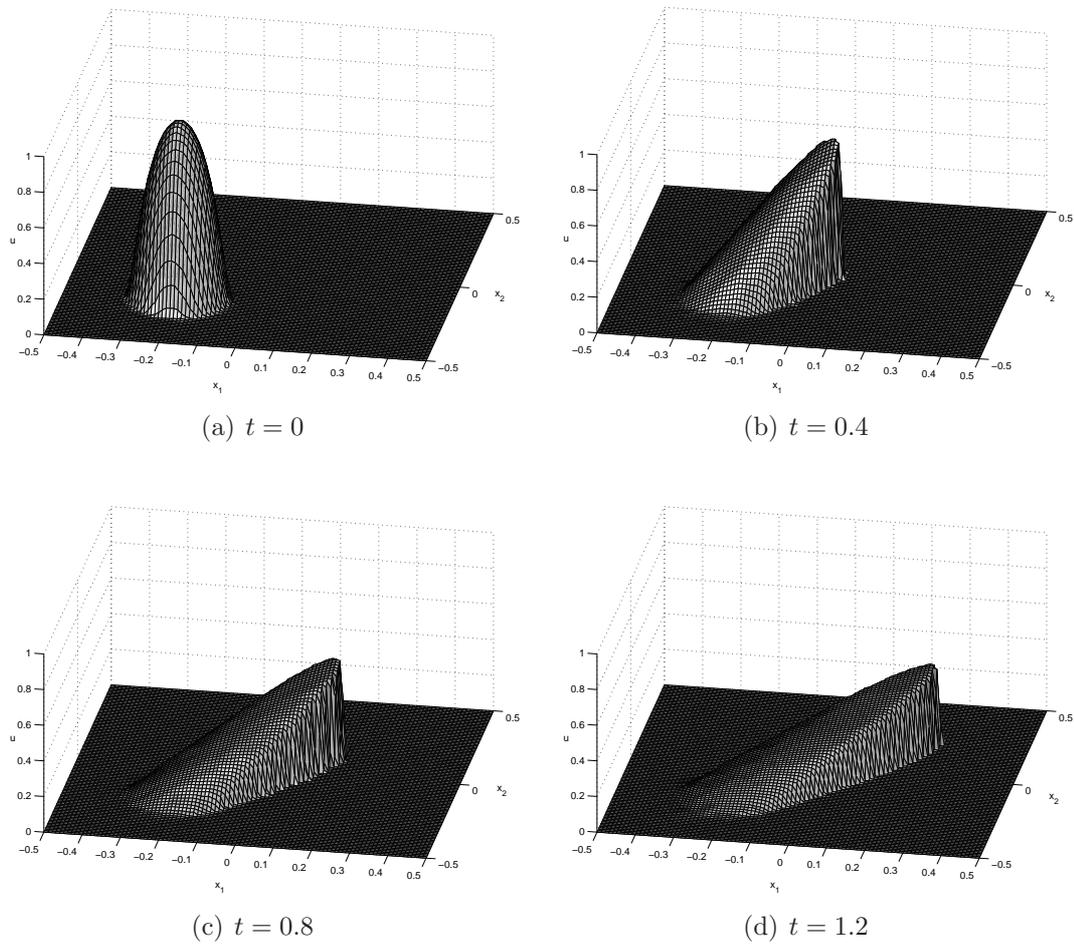


Figure 5.6: Solution of Burgers' equation at times (a) $t = 0$, (b) $t = 0.4$, (c) $t = 0.8$, and (d) $t = 1.2$ using the ADER-WENO method.

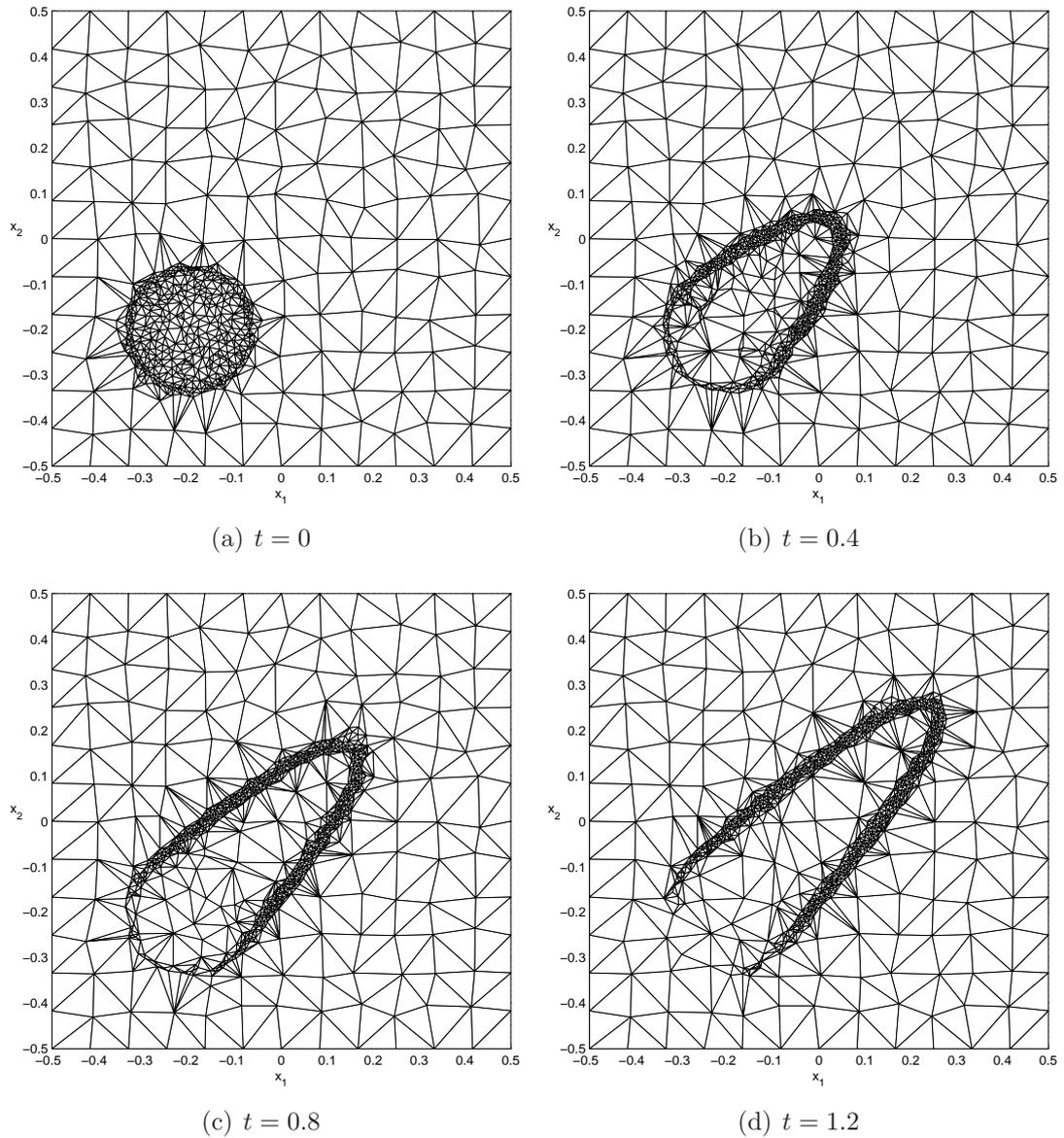


Figure 5.7: Adapted mesh for the solution of Burgers' equation at times (a) $t = 0$, (b) $t = 0.4$, (c) $t = 0.8$, and (d) $t = 1.2$ using the ADER-WENO method.

of cells after some time. This is because the support of the solution grows with time and this leads to a growth in the length of the shock front. However, $\phi_{2,3}(r) = r^4 \log(r)$ uses fewer cells, which may be a case for the use of higher order methods. We notice that for the RK-WENO method with $\phi_1(r) = r$ reconstruction, there is initially a very large increase in the number of cells. We think this is because during the early stages of the simulation it does not coarsen the mesh effectively in the regions of the rarefaction wave. Furthermore, since it is a low order method, the use of coarser meshes will lead to a decrease in accuracy.

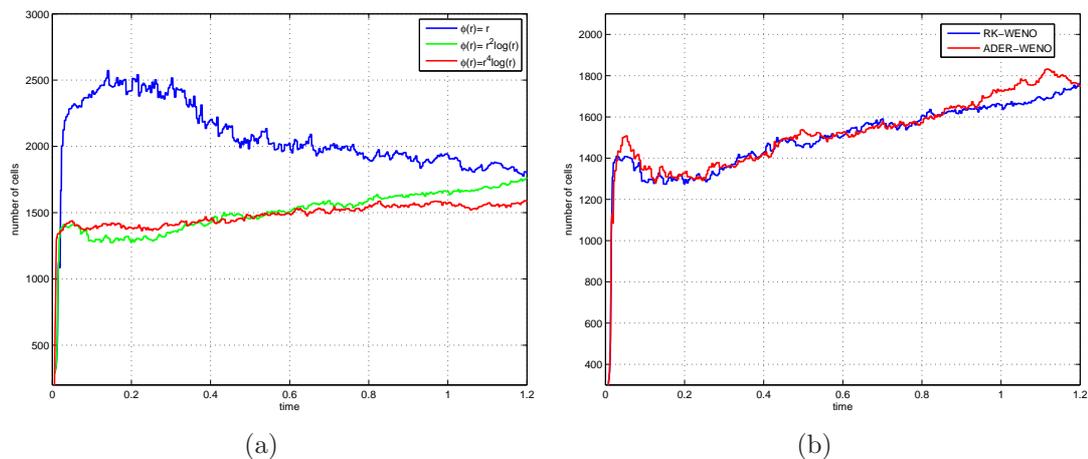


Figure 5.8: (a) Number of cells for different WENO reconstructions and (b) the number of cells used in the RK-WENO and ADER-WENO methods with thin plate spline reconstruction.

Figure 5.8(b) is a comparison of the number of cells when the simulation is done by the RK-WENO and ADER-WENO methods with thin plate spline reconstruction and the same number of initial mesh refinement cycles. We observe that throughout the simulation, the RK-WENO and ADER-WENO methods use approximately the same number of cells.

5.4.2 Time dependent velocity fields: The swirling deformation problem

While it is important to study the adaptive solution of nonlinear conservation laws where shocks may develop in finite time, the resolution of discontinuities for variable-coefficient linear conservation laws is sometimes a numerically challenging task.

Consider the linear model problem in which the advection of the scalar function $u(t, \mathbf{x})$ in a specified time dependent velocity field $\sigma(t, \mathbf{x}) = (\sigma_1(t, \mathbf{x}), \sigma_2(t, \mathbf{x}))$ is modelled by the conservation law:

$$u_t + (\sigma_1(t, \mathbf{x})u)_{x_1} + (\sigma_2(t, \mathbf{x})u)_{x_2} = 0, \quad (5.9)$$

with initial condition

$$u(0, \mathbf{x}) = u_0(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^2. \quad (5.10)$$

If we assume that the flow is divergence-free, so that

$$(\sigma_1(t, \mathbf{x}))_{x_1} + (\sigma_2(t, \mathbf{x}))_{x_2} = 0,$$

everywhere, then (5.9), (5.10) can be written equivalently as a variable coefficient advection equation

$$u_t + \sigma_1(t, \mathbf{x})u_{x_1} + \sigma_2(t, \mathbf{x})u_{x_2} = 0, \quad (5.11)$$

$$u(0, \mathbf{x}) = u_0(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^2. \quad (5.12)$$

In this subsection, we look at a useful numerical test due to Leveque [77] for (5.11). It uses a swirling deformation flow of the form

$$\sigma_1(t, \mathbf{x}) = \sin^2(\pi x_1) \sin(2\pi x_2)q(t) \quad \text{and} \quad \sigma_2(t, \mathbf{x}) = -\sin^2(\pi x_2) \sin(2\pi x_1)q(t)$$

on the unit square $[0, 1]^2$. This velocity field satisfies $\sigma_1(\mathbf{x}) = \sigma_2(\mathbf{x}) = 0$ on the boundaries of the unit square. The time dependence is introduced into the velocity field by using the function $q(t)$ which we define as

$$q(t) = \cos\left(\frac{\pi t}{T}\right)$$

on the time interval $[0, T]$. The initial condition is a three-body structure on the unit square as shown in Figure 5.9. It consists of a smooth hump, a cone and a slotted cylinder like the one used by Zalesak [148]. The smooth hump has the form

$$g(\mathbf{x}) = \frac{1}{2}(1 + \cos(\pi r(\mathbf{x}))),$$

where

$$r(\mathbf{x}) = \min\left(\sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2}, r_0\right) / r_0.$$

In this test case, $c_1 = 0.25$, $c_2 = 0.5$, and $r_0 = 0.15$. The slotted cylinder and cone have radii 0.15 and centres at $(0.5, 0.75)$ and $(0.5, 0.25)$ respectively. The cylinder has a slot of width 0.06 and length 0.22.

When solving this problem with many advection schemes, the steep gradients may be poorly resolved from initially well-resolved fields due to the stretching and deformation caused by the velocity field. Figure 5.10(a) shows the velocity field for time $0 \leq t \leq T/2$ while Figure 5.10(b) shows the velocity field for time $T/2 < t \leq T$. Thus, the initially well-resolved initial condition deforms into three narrow crescents before reversing di-

rection and returning back to its original shape and position. During simulation, the flow slows down and reverses direction in such a way that the initial shape should be recovered at time T , i.e. $u(T, \mathbf{x}) = u(0, \mathbf{x})$. This is a very practical test problem since we know the true solution at time T even though the flow field has quite a complicated structure.

To ensure that the solution is well resolved throughout the simulation we solve this problem with an adaptive RK-WENO method with thin plate spline reconstruction. Here we use $T = 1.5$ and so at time $T/2$ the initial data is most deformed. Figures 5.11, 5.12 and 5.13 show the adapted meshes, the filled contour plots and the 3D plots respectively of the solution at time $t = 0.375$, $t = 0.75$, $t = 1.125$, and $t = 1.5$.

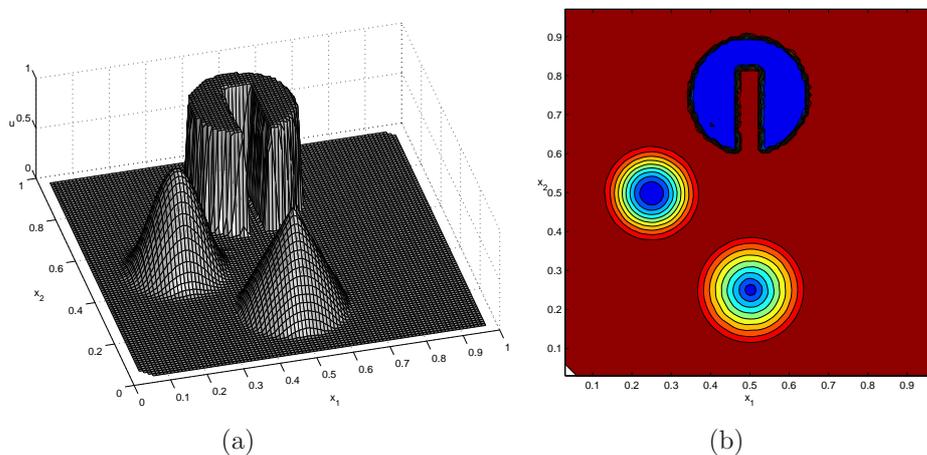


Figure 5.9: (a) The 3D plot and (b) the filled contour plot for initial condition for the swirling deformation problem.

We observe that the initial shapes have been recovered fairly successfully. Our numerical solution is nonoscillatory but some numerical diffusion is witnessed. We also observe the shape-preserving property of the method, cf. Figures 5.9(b) and 5.12(d) or Figures 5.9(a) and 5.13(d). Finally, from Figure 5.14, we see that the number of cells increases as the initial condition becomes more deformed. We observe the maximum number of cells at about $T/2$, where the initial profile is most deformed after which the number of cells decreases steadily. By the end of the simulation, it reduces to nearly the level it was after the initial mesh refinement. This example again confirms the robustness of the adaptive algorithm.

5.4.3 Two-phase flow in porous media

The mathematical modelling of two-phase flow has applications in several branches of science and engineering. For example, in petroleum engineering, one may wish to look at the modelling of an oil reservoir: one technique involves modelling an enhanced oil

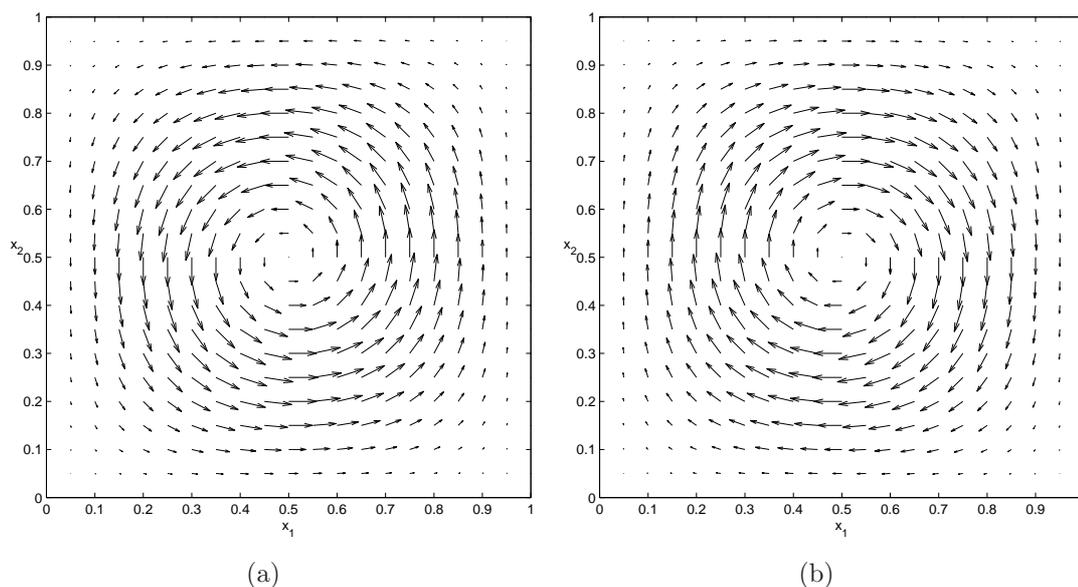


Figure 5.10: (a) Velocity field for time $0 \leq t \leq T/2$ and (b) for time $T/2 < t \leq T$ for the swirling deformation problem.

recovery process where a wetting fluid, say water, is injected into an oil-saturated porous medium in order to displace the oil (the non-wetting fluid) so that it can be collected.

In this subsection, we are interested in oil-water simulations within a homogeneous porous medium. We provide a detailed description of the model used here in Appendix C. It is based on the book of Peaceman [93] and on [64, 66].

The two-phase flow of two immiscible, incompressible fluids through a homogeneous porous medium in the absence of capillary pressure and gravitational effects is modelled by the following three equations:

- The *Buckley-Leverett* equation [17]

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla F(u) = 0, \quad (5.13)$$

with

$$F(u) = \frac{u^2}{u^2 + m(1-u)^2} \quad (5.14)$$

where \mathbf{v} is the *velocity field* and m is the ratio of the viscosities of the two fluids. This means $m = \mu_w/\mu_n$ where μ_w and μ_n are the water and oil phase viscosities.

- The *incompressibility relation*

$$\nabla \cdot \mathbf{v}(t, \mathbf{x}) = 0. \quad (5.15)$$

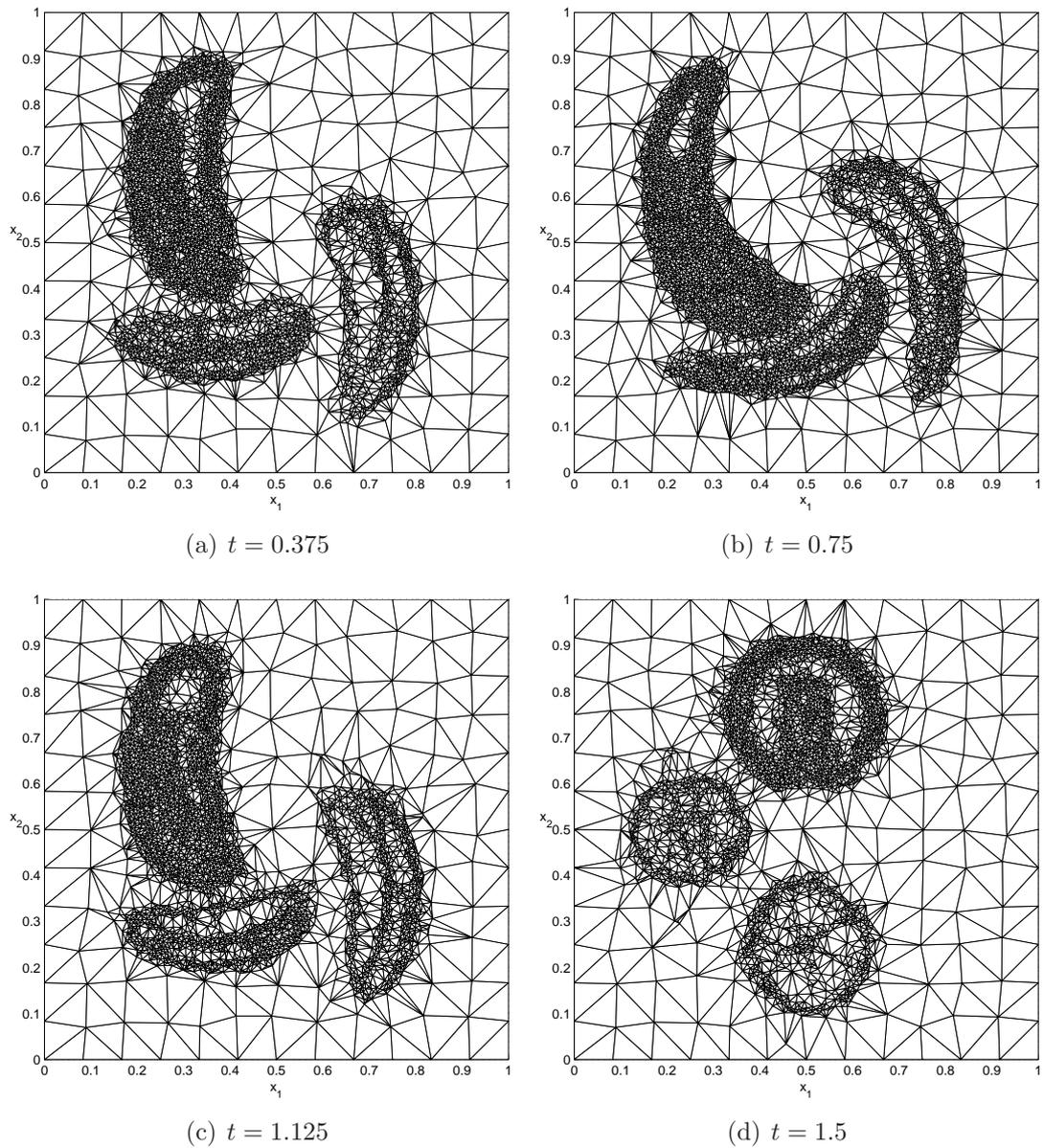


Figure 5.11: Adapted mesh for the solution of the swirling deformation problem at times (a) $t = 0.375$, (b) $t = 0.75$, (c) $t = 1.125$, and (d) $t = 1.5$.

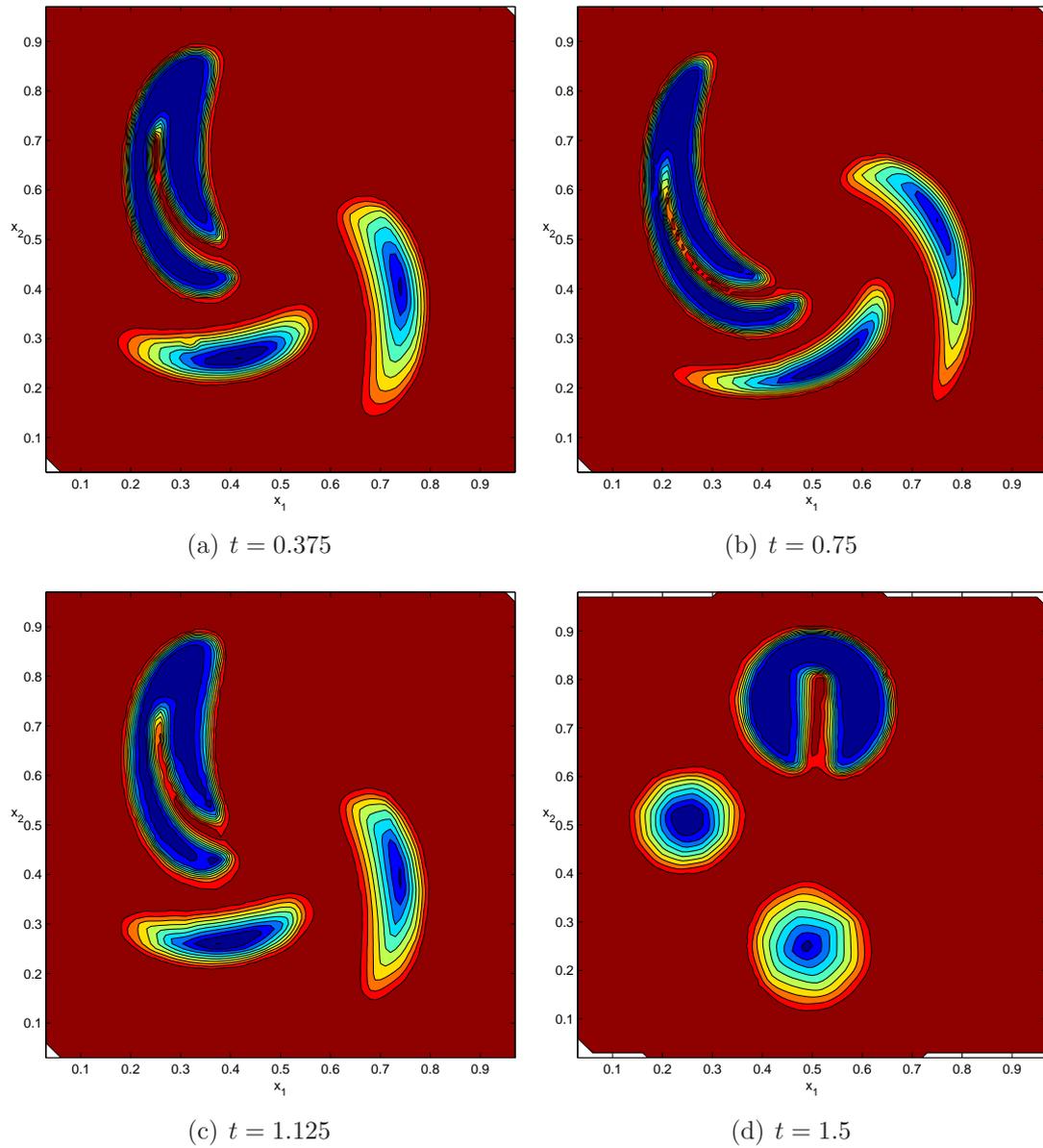


Figure 5.12: Filled contour plots of the solution of the swirling deformation problem at times (a) $t = 0.375$, (b) $t = 0.75$, (c) $t = 1.125$, and (d) $t = 1.5$.

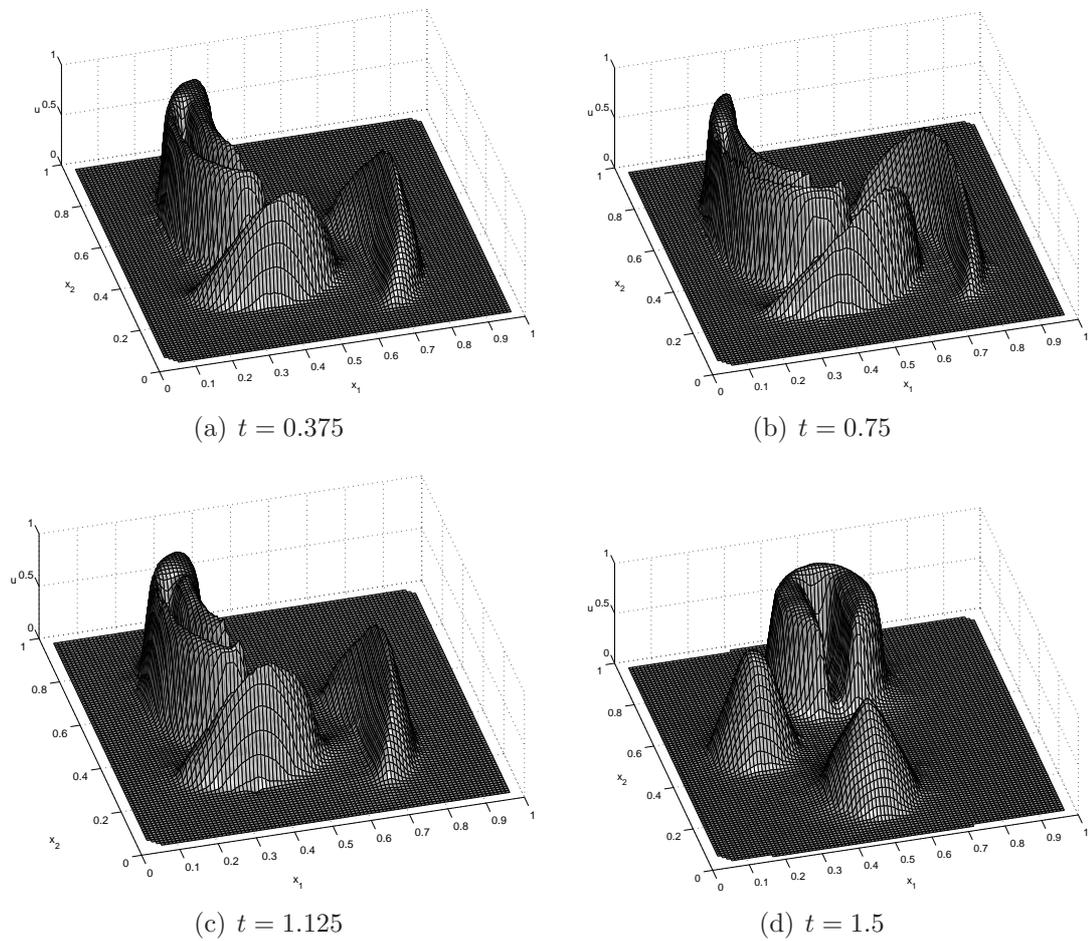


Figure 5.13: 3D plots of the solution of the swirling deformation problem at times (a) $t = 0.375$, (b) $t = 0.75$, (c) $t = 1.125$, and (d) $t = 1.5$.

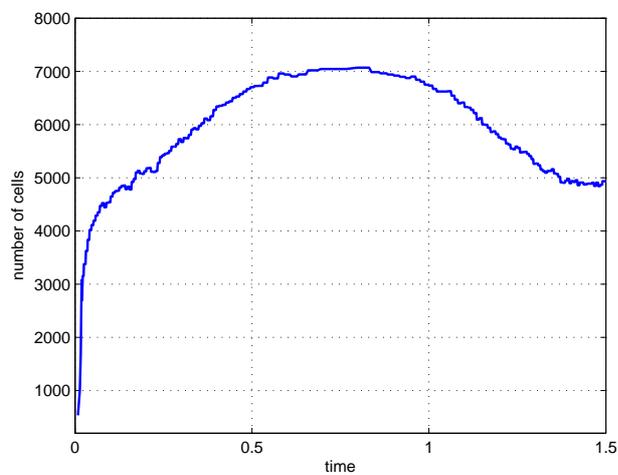


Figure 5.14: Number of cells during the simulation of the swirling deformation problem.

- *Darcy's law*

$$\mathbf{v} = -M_t(u)\nabla\mathbf{p}(t, \mathbf{x}), \quad (5.16)$$

where \mathbf{p} denotes the *pressure* and the *total mobility*

$$M_t(u) = \frac{u^2}{\mu_w} + \frac{(1-u)^2}{\mu_n} \quad (5.17)$$

depends on the *permeability* of the medium and on the viscosity ratio m .

The solution u of (5.13), (5.15), (5.16) is the saturation of the wetting fluid (water) in the non-wetting fluid (oil). Therefore, the value $u(t, \mathbf{x})$ is, at a point \mathbf{x} and time t , the fraction of available volume filled with water.

The five-spot problem

The five-spot problem is a classical model problem from oil reservoir engineering. The design of the model is such that the reservoir is a square, in which water is injected at the centre and oil recovered at the corners.

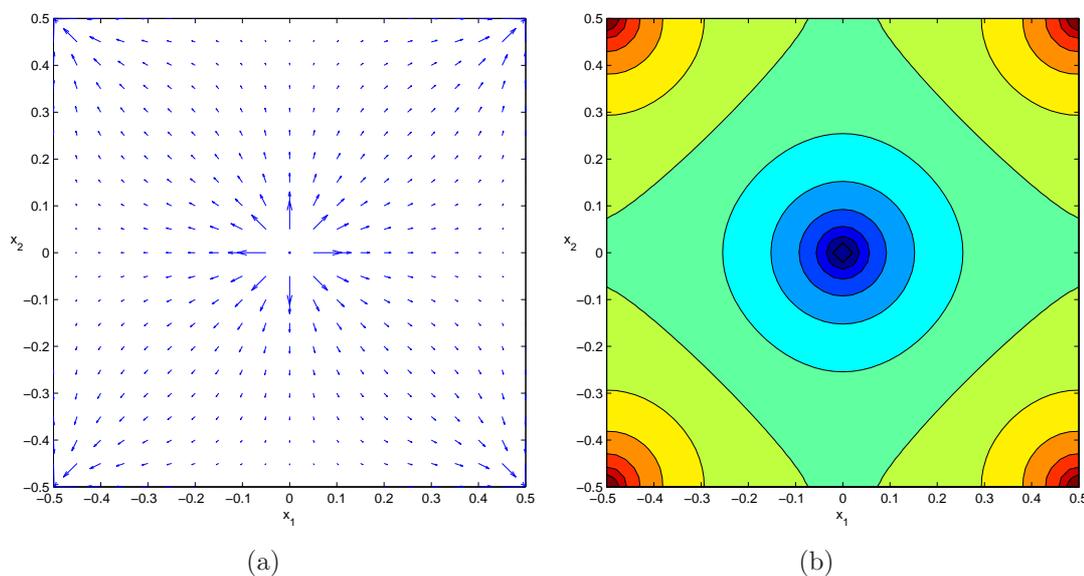


Figure 5.15: (a) Velocity field and (b) filled contours of the pressure field for the five spot problem.

In this section, we assume our computational domain $\Omega = [-0.5, 0.5]^2$ is a bounded oil reservoir. The pores of the oil reservoir are filled with the non-wetting fluid (oil, $u \equiv 0$), before the wetting fluid (water, $u \equiv 1$) is injected through a single injection well, placed at the centre $\alpha = (0, 0)$ of the computational domain Ω . As the simulation advances in time, the oil is displaced by the water towards the four corner points

$$\mathcal{C} = \{(-0.5, -0.5), (-0.5, 0.5), (0.5, -0.5), (0.5, 0.5)\}$$

of the computational domain Ω .

We solve the five-spot problem by computing a numerical solution of (5.13), (5.15), and (5.16) on Ω with initial data

$$u_0(\mathbf{x}) = \begin{cases} 1, & \|\mathbf{x} - \alpha\| < R; \\ 0, & \text{otherwise,} \end{cases} \quad (5.18)$$

where $R = 0.02$ is the radius of the injection well at the centre $\alpha \in \Omega$ for $t \in [0, T]$. This initial condition illustrates a situation of pure water being injected into an initially saturated oil reservoir.

A few assumptions can be made to simplify the five-spot problem. We first assume unit mobility, $M_t \equiv 1$, so that the elliptic equations (5.15) and (5.16) decouple from the Buckley-Leverett equation, which is hyperbolic. Furthermore, we use a stationary pressure field, i.e.

$$\mathbf{p}(\mathbf{x}) \equiv \mathbf{p}(t, \mathbf{x}) = \sum_{\beta \in \mathcal{C}} \log(\|\mathbf{x} - \beta\|) - \log(\|\mathbf{x} - \alpha\|), \quad \text{for all } \mathbf{x} \in \Omega, \quad t \in [0, T]$$

which altogether gives us, from (5.16), a steady velocity field

$$\mathbf{v} = -\nabla \mathbf{p}, \quad (5.19)$$

which satisfies (5.15). These assumptions mean that we can solve the five spot problem by a numerical simulation of (5.13).

Figure 5.15 shows the velocity vectors and the filled contour plot of the pressure field \mathbf{p} . We note that the pressure \mathbf{p} has singularities at the four corner of Ω and at the centre α . This leads to high velocity near the five wells, but small velocity between the wells as reflected in Figure 5.15(b).

The nature and location of the shock front at the interface of the two phases is of great importance in the simulation of oil reservoirs. Therefore, the accurate resolution of the shock front must be treated with care. To this end, we will use the adaptive ADER-WENO method with thin plate spline reconstruction to solve the simplified version of the five-spot problem. The adaptive mesh on which the solution was computed, and the advancement of the displacement of oil by water are shown in Figures 5.16 and 5.17 respectively for the ADER-WENO method. The plots are shown at six different times: t_0 , t_{490} , t_{980} , t_{1470} , t_{1960} and t_{2450} .

We note that the water saturation u lies between zero and one; $u \equiv 1$ means we have pure water and $u \equiv 0$ means we have pure oil. We use $m = 0.5$ as a water-oil viscosity ratio in our simulation. We also permit outflow boundaries so that the oil can leave the computational domain.

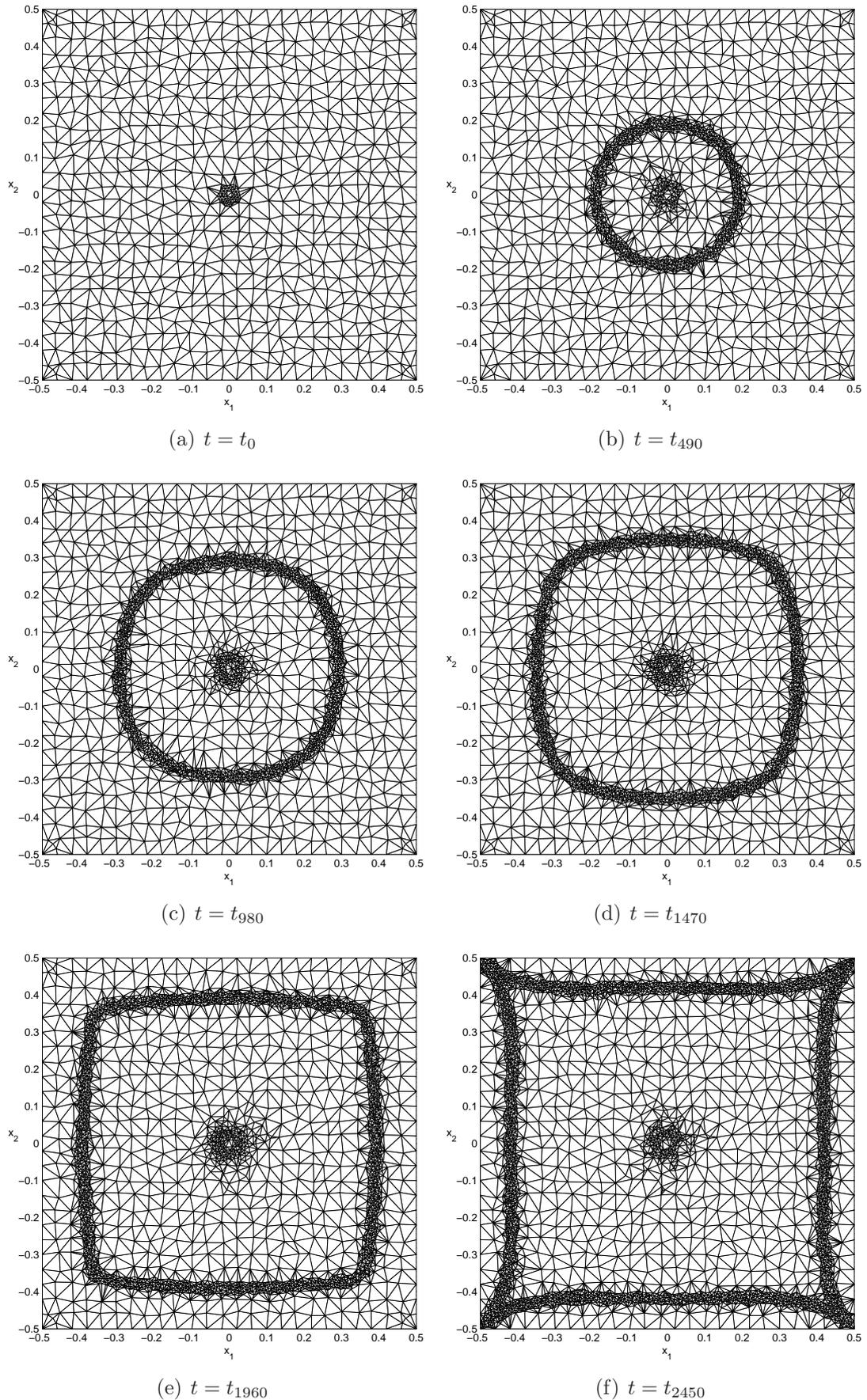


Figure 5.16: Adapted mesh for the solution of the five spot problem by the ADER-WENO method at six different times (a) $t = t_0$, (b) $t = t_{490}$, (c) $t = t_{980}$, (d) $t = t_{1470}$, (e) $t = t_{1960}$, and (f) $t = t_{2450}$.

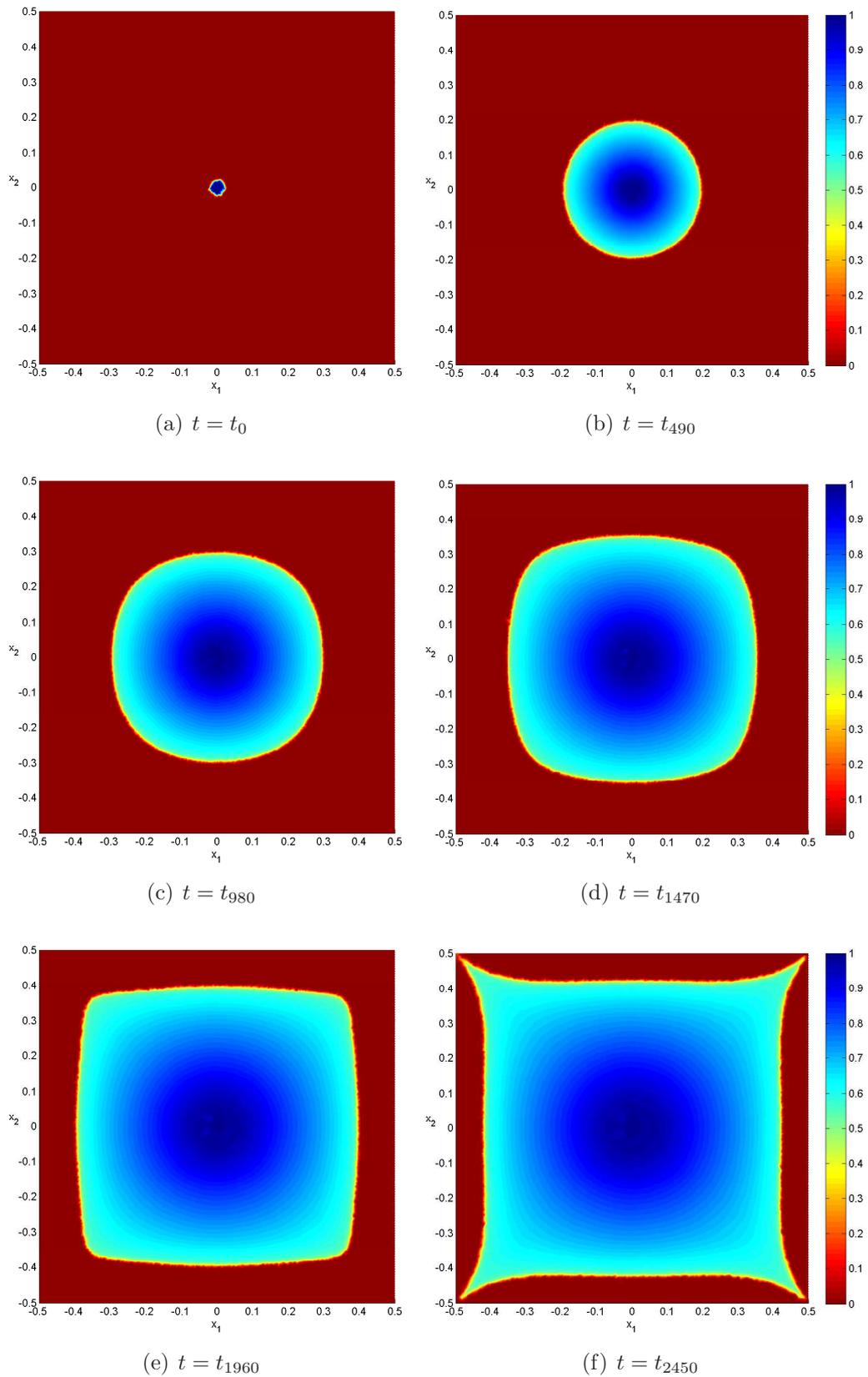


Figure 5.17: Colour plots showing the water saturation u of the five spot problem at six different times (a) $t = t_0$, (b) $t = t_{490}$, (c) $t = t_{980}$, (d) $t = t_{1470}$, (e) $t = t_{1960}$, and (f) $t = t_{2450}$.

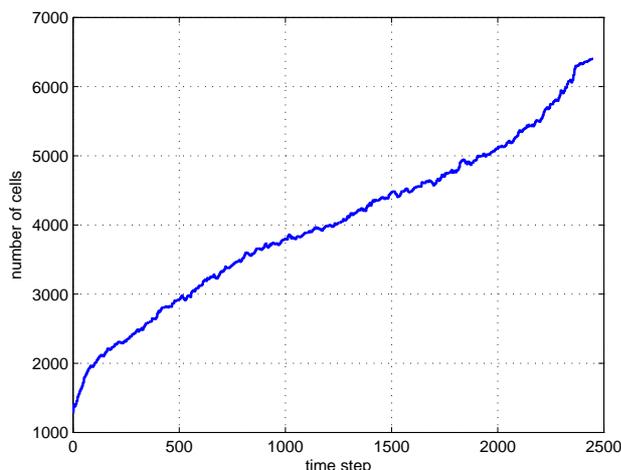


Figure 5.18: Number of cells during the simulation of the five spot problem.

Immediately water is pumped into the well, a shock is formed as we can see in Figure 5.17(a). This shock, which is at the interface of the pure oil and a mixture of oil and water, moves from the centre of our model reservoir towards the four production wells at the corners \mathcal{C} of the computational domain Ω . This means that during the simulation, the oil is displaced by the water. Note that the time at which the shock front arrives at the production well is known as the *breakthrough time*.

Due to the adaptive mesh refinement, (see Figures 5.16), the shock front propagation of the solution is captured very well. We also notice that the mesh stays refined in the neighbourhood of the injection well. Furthermore, once the shock passes a particular location, our adaptive strategy coarsens/derefiners the mesh behind it to its original level. This indeed reduces the computational cost while at the same time improving the accuracy of the method. This again confirms the utility of the adaptive strategy used in this work.

Figure 5.17 demonstrates how the diameter of the shock front increases linearly with time and we observe from Figure 5.18 that the number of triangles is approximately directly proportional to the diameter of the shock front.

In conclusion, we see that our adaptive method with thin plate spline WENO reconstruction is able to effectively resolve the moving sharp shock fronts with the help of local mesh refinement. We also see the non-oscillatory nature of the scheme: no spurious oscillations are observed. Furthermore, the adaptive methods allow us to reduce the number of required cells, and hence the computational cost, as compared to any method that uses a fixed computational mesh.

5.5 Mesh & Stencil Adaptivity

We seek further reduction in computational cost by combining mesh adaptivity with stencil adaptivity for the RK-WENO method using thin plate spline reconstruction. To accomplish this, we use the same indicator (5.2) for both mesh adaptivity and stencil adaptivity. Given the threshold values ϑ , ϑ_r and ϑ_d , we will use stencils of size four for reconstruction on a cell T if $\varepsilon_T \leq \vartheta \cdot \varepsilon^*$ and stencils of size seven if $\varepsilon_T > \vartheta \cdot \varepsilon^*$ where we set $\vartheta = 0.035$. We will employ Definition 5.2 for mesh adaptivity where we set $\vartheta_r = 0.05$ and $\vartheta_d = 0.01$. From our numerical tests, we observed that using $\vartheta < \vartheta_r$ gave better results. We demonstrate the benefits of this combination by solving the linear advection

method	E_1	E_2	E_∞	N_4	N_7
mesh adaptivity	$2.9060 \cdot 10^{-3}$	$1.0821 \cdot 10^{-2}$	$8.3975 \cdot 10^{-2}$	-	1480
mesh & stencil adaptivity	$2.8926 \cdot 10^{-3}$	$1.0618 \cdot 10^{-2}$	$8.2410 \cdot 10^{-2}$	1018	474

Table 5.3: Comparing mesh adaptivity and mesh & stencil adaptivity for the linear advection equation.

equation (3.35) with initial data (5.3) and show the results at $t = 0.25$ for both mesh adaptivity and mesh & stencil adaptivity in Table 5.3. At the end of our simulation, 1480 cells were used for mesh adaptivity and 1492 cells were used for mesh & stencil adaptivity. We noticed a reduction in computational time for mesh & stencil adaptivity because reconstruction with stencils of size seven was only applied to 474 cells. The

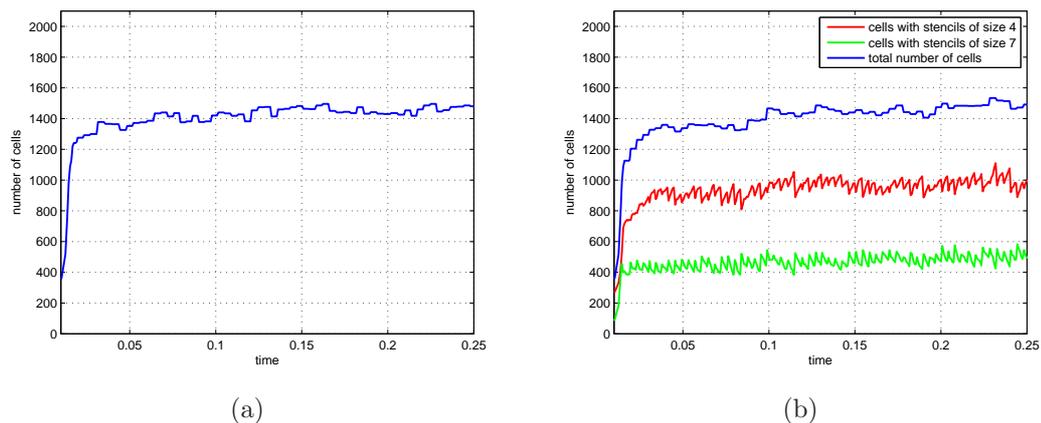


Figure 5.19: (a) Number of cells for mesh adaptivity and (b) number of cells and stencil size distribution for mesh & stencil adaptivity for the linear advection equation.

number of cells used in the simulation along with the distribution of stencil sizes for mesh & stencil adaptivity are shown in Figure 5.19(b).

We will also use the mesh & stencil adaptivity to solve Burgers' equation (3.37) with initial data (5.3) and display the results at $t = 1.2$ for both mesh adaptivity and mesh

& stencil adaptivity in Table 5.4.

method	E_1	E_2	E_∞	N_4	N_7
mesh adaptivity	$1.6848 \cdot 10^{-3}$	$1.1745 \cdot 10^{-2}$	$7.8050 \cdot 10^{-2}$	-	1762
mesh & stencil adaptivity	$1.6702 \cdot 10^{-3}$	$1.1152 \cdot 10^{-2}$	$7.7832 \cdot 10^{-2}$	1112	574

Table 5.4: Comparing mesh adaptivity and mesh & stencil adaptivity for Burgers' equation.

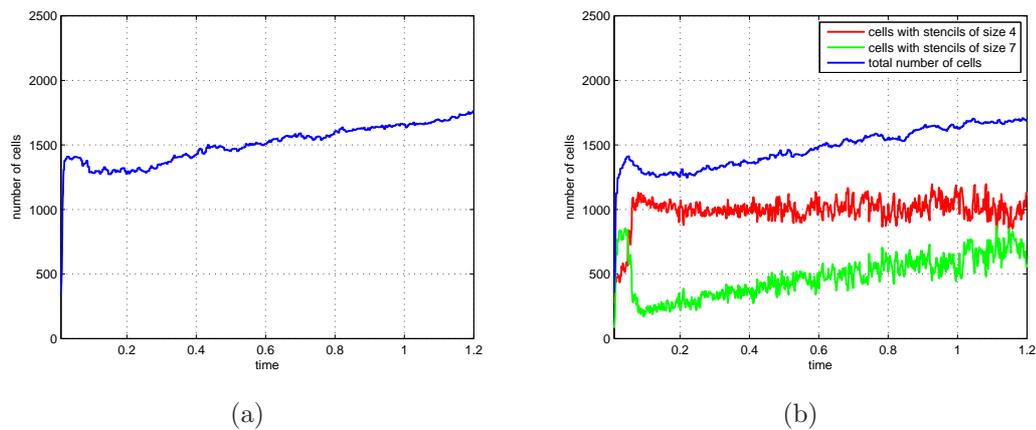


Figure 5.20: (a) Number of cells for mesh adaptivity and (b) number of cells and stencil size distribution for mesh & stencil adaptivity for Burgers' equation.

The number of cells used in the simulation along with the distribution of stencil sizes for mesh & stencil adaptivity are shown in Figure 5.20(b).

At the end of the simulation, 1762 cells were used for mesh adaptivity while just 1686 cells were used for mesh & stencil adaptivity. We noticed a reduction in computational time when mesh & stencil adaptivity is implemented because reconstruction with stencils of size seven was only applied to 574 cells. A significant advantage of the mesh & stencil adaptivity lies in the fact that the stencil adaptivity step does not require any extra computation of the error indicators since the same error indicator is used for both the mesh adaptivity and stencil adaptivity steps. Finally, as displayed in Table 5.4, there is little difference in the errors and so for the sake of computational cost, we believe that coupling mesh adaptivity and stencil adaptivity is very useful in practice.

Chapter 6

Mesh & Order Adaptivity

Most adaptive methods for PDEs, including the methods presented in Chapter 5, simply refine the mesh in regions where the error indicator is large and coarsen the mesh where the error indicator is small while keeping the order of the recovery function used in the spatial discretisation fixed.

However, if the solution of the governing PDE is smooth in large portions of the computational domain (as is usually the case in many physically relevant situations), then using a mesh-only adaptive strategy in combination with a fixed low order recovery function may not be very efficient.

Moreover, since high order methods usually degenerate to low order across discontinuities, the use of high order recovery functions, which are computationally more expensive, may not be the optimal choice near discontinuities. This observation motivated the design and analysis *hp*-finite element methods which combine mesh refinement/derefinement with order variation procedures. The theory and implementation of *hp*-methods have been treated in several books including Karniadakis & Sherwin [71], Schwab [104], and Szabo & Babuška [121]. Generally speaking, *p*-refinement (order variation) is more efficient in regions where the solution of the governing equation is smooth while *h*-refinement (mesh adaptivity) is more suitable near singularities and non-resolved steep gradients.

The utility of *hp*-FEMs has been well established for elliptic and parabolic PDEs, see [4, 58]. They are generally known to offer greater flexibility and improved efficiency when compared to mesh refinement methods. The application of *hp*-FEMs to hyperbolic problems is less standard although some work has been done in this direction. In the thesis of Bey [14] and in the work of Houston *et al* [40, 57, 71], the *hp*-FEM was used to solve steady hyperbolic problems. Results for time dependent conservation laws combined with Runge-Kutta time stepping can be found in the work of Devine & Flaherty [31] and Dedner & Oehlberger [30]. All the work on *hp*-FEMs for steady and unsteady conservation laws show their suitability and effectiveness for the numerical

solution of this class of problems. A good reason for using *hp*-methods in the numerical solution of hyperbolic problems lies in the fact that although the solution may exhibit local discontinuities, in large parts of the computational domain, the solution is smooth.

To the best of our knowledge, almost all of the adaptive strategies for finite volume methods have been based on either mesh movement (*r*-refinement), e.g. Tang & Tang [123] and Baines *et al* [7], or mesh refinement/coarsening (*h*-refinement) e.g. Smit *et al* [112] and Käser & Iske [73]. On the other hand, Kurganov *et al* [75] implemented a *scheme adaptation* algorithm in combination with central finite volume schemes. They used a low order slope limiting method in regions where the solution is rough and a high order reconstruction (like the fifth order WENO reconstruction) where the solution is smooth. Thus, they performed a sort of *p*-adaptivity with just two possible choices of recovery functions. The local smoothness was determined by a smoothness indicator computed from the weak local truncation error in conjunction with cubic B-splines. The same smoothness indicator was also used to separately implement a mesh adaptation procedure.

Our main goal in this chapter is to see if the desirable properties of *hp*-adaptivity in FEMs can be used as a basis to construct an adaptive finite volume method that incorporates both mesh refinement and order variation procedures: we call this *mesh & order adaptivity*. The method we seek to develop is based primarily on the error indicator (5.2), the paper of Devine & Flaherty [31] and the work of Houston *et al* [40, 57].

The order variation procedure used here is not based on changing the degree of the polynomial as in *hp*-FEMs, but on varying the order of the radial recovery function. We have three basis functions available to us: the RBF interpolants with (i) $\phi_1(r) = r$, (ii) $\phi_{2,2}(r) = r^2 \log(r)$ and (iii) $\phi_{2,3}(r) = r^4 \log(r)$.

For mesh refinement/coarsening, we use the *newest vertex bisection* method which was first proposed by Sewell [106]. The version of the method we use here is based on the work of Chen & Zhang [20, 21, 22] who implemented the newest vertex bisection for the refinement and coarsening on triangular meshes in MATLAB. We present a summary of the work of Chen & Zhang in Appendix B. We use this mesh refinement strategy, which is different from the one in Chapter 5, because we need to have control of the cells in the mesh and the associated data structures as well as keep a record of the history of subdivisions. The aim of this strategy is to refine conforming triangulations by dividing *triangles* into *subtriangles*, the so-called *children* of a common *father*, and to coarsen such triangulations by unifying the children of a common father, the so-called *brothers*. The father is restored by the unification of the children. It is therefore possible to coarsen a refined triangulation up to its initial state.

6.1 The Refinement Strategy

In recent years, there has been an explosion of research on the theory, design and implementation of hp -methods. In particular, it has been shown that with a proper combination of local h and p refinement, exponential convergence could be achieved even in the presence of singularities [104].

A key feature in any hp -algorithm is the decision on each element $T \in \mathcal{T}$ as to which strategy (i.e. mesh refinement/derefinement or order variation) to use in order to achieve the greatest reduction in error in relation to computational cost.

In contrast to mesh-only adaptive methods, a mesh & order adaptive strategy must have two essential ingredients [57]:

1. an error estimation procedure,
2. a steering criterion for deciding whether to refine or coarsen an element or to increase or decrease the order of reconstruction.

Based on the size of the local error indicator (5.2), for each triangle $T \in \mathcal{T}$, our algorithm uses an appropriate criterion to determine elements in high error regions which are to be refined and those in low error regions which are to be derefined. In this chapter, we still use the strategy presented in Definition 5.2.

Once an element $T \in \mathcal{T}$ has been marked for refinement or derefinement, a decision has to be made whether the local mesh size h or the order of the reconstruction RBF should be varied accordingly. The choice of whether to perform either mesh refinement/coarsening or order refinement/coarsening usually depends on the local smoothness of the solution [5, 57]. The method we propose for determining the local smoothness of the solution is described in the next section.

We first treat the case where an element has been marked for refinement. If u is locally smooth, then order enrichment will be more effective than mesh refinement, since the error is supposed to reduce rapidly within the current element T as the order is increased. In this way, we benefit from the higher convergence rate of high-order methods. On the other hand, if u has low regularity within an element T , then mesh refinement will be performed to improve accuracy. In regions where the solution is non-smooth, raising the order of reconstruction usually yields no benefit.

If an element has been marked for derefinement, then the strategy is to coarsen the mesh where the solution u is smooth and decrease the order of the reconstruction RBF if u is not smooth [4, 40].

6.2 Estimation of Smoothness

In this section, we are concerned with determining whether the solution is locally smooth on each element $T \in \mathcal{T}$. To address this issue, we first of all recall the theorem of Wendland [143] which we presented in Section 2.2. The theorem implied that for the interpolant with the polyharmonic spline $\phi_{d,k}$, we have

$$\|u - s\|_{L_q(\Omega)} \leq Ch^{k-d(1/2-1/q)_+} |u|_{\text{BL}_k(\Omega)}.$$

In addition, for interpolation with $\phi_1(r) = r$, we recall from Theorem ?? that

$$|u - s| \leq Ch^{\beta/2} |u|_{\mathcal{N}_\phi}.$$

Now, suppose s_1 , s_2 and s_3 denote the WENO interpolants corresponding to the basis functions $\phi_1(r) = r$, $\phi_{2,2}(r) = r^2 \log(r)$ and $\phi_{2,3}(r) = r^4 \log(r)$ respectively on a cell T , then we propose the following measure of smoothness.

If

$$\frac{\|u - s_2\|_{L_2}}{\|u - s_1\|_{L_2}} \geq c_1 h$$

then we expect the solution to be smooth on the basis of the known approximation theory. Thus, if the element is marked for refinement, we increase the order. Moreover, if

$$\frac{\|u - s_3\|_{L_2}}{\|u - s_2\|_{L_2}} \geq c_2 h$$

we further increase the order of the reconstruction. If the solution is non-smooth, we will subdivide the cell.

On the other hand, in low error regions, if

$$\frac{\|u - s_2\|_{L_2}}{\|u - s_1\|_{L_2}} \geq c_1 h,$$

we expect the solution to be smooth and if the element is marked for derefinement, we coarsen the mesh. If the element that is marked for derefinement lies in a non-smooth region of the solution; we decrease the order of reconstruction.

6.3 The Algorithm

The mesh & order adaptive algorithm is described below. When enrichment is necessary, we first enrich the order of approximation on high-error cells in smooth regions by replacing $\phi_1(r) = r$ by $\phi_{2,2}(r) = r^2 \log(r)$. Enrichment is repeated until no further order enrichment can be done. For higher order cells in non-smooth regions, we subdivide

the triangles. In low error regions, we either coarsen the mesh or reduce the order of reconstruction depending on the local smoothness. At each time step, several mesh and order refinement cycles can be performed.

In summary, the mesh & order algorithm for advancing one step in time from t^n to t^{n+1} is given as follows.

Algorithm 6.1 (Mesh & Order adaptivity)

Input: *Triangulation \mathcal{T} with cell averages \bar{u}_T^n , $T \in \mathcal{T}$, time step $\Delta t > 0$ and threshold values ϑ_r and ϑ_d .*

- *Calculate the RK-WENO solution based on an initially refined mesh and $\phi_1(r) = r$.*
- **FOR** *each $T \in \mathcal{T}$ DO*
 - (1) *Calculate the error indicators ε_T for each element $T \in \mathcal{T}$ and compute $\varepsilon^* = \max_{T \in \mathcal{T}} \varepsilon_T$.*
 - (2) *If $\varepsilon_T \geq \vartheta_r \cdot \varepsilon^*$, mark T for refinement and if $\varepsilon_T \leq \vartheta_d \cdot \varepsilon^*$, mark T for de-refinement.*
 - (3) *If T is marked for refinement, decide whether to perform mesh refinement or order enrichment. If the solution is smooth, increase the order and if non-smooth, refine T and keep the RBF the same on the resulting sub-elements.*
 - (4) *If T is marked for coarsening, decide whether to perform mesh de-refinement or order reduction. If the solution is smooth, coarsen the element and if non-smooth, reduce the order of reconstruction.*
 - (5) *Compute the numerical flux using (3.14).*
 - (6) *Form the numerical scheme (3.15).*

Output: *Updated cell average values \bar{u}_T^{n+1} for all $T \in \mathcal{T}$.*

Other *hp*-adaptive strategies for hyperbolic problems include the *3-step strategy* used in the thesis of Bey [14]. The goal of this method is to deliver a solution with a specified

error in only three steps. The first step involves the construction of an initial hp -mesh as well as setting the required parameters. In the second step, the polynomial degree is kept fixed while the mesh is refined until the error is reduced to a specified level. In the third step, the mesh is kept fixed while the polynomial degree is increased until the error is further reduced to a specified level. Dedner and Ohlberger [30] use a projection operator for p -adaptivity and an error indicator based on element, projection and jump residuals for the mesh adaptivity.

There are other methods used in the literature for determining which elements are to be marked for refinement/derefinement on the basis of the error indicator. For example, in [40] and [58], the fixed fraction strategy was used with refinement and derefinement fractions set to 20% and 10% respectively. Devine & Flaherty [31] mark the elements where the error indicator is greater than a user specified tolerance for refinement and those less than a specified tolerance for derefinement.

For determining the local smoothness, Houston *et al* [57] extended the method of Ainsworth and Senior [5] to hyperbolic problems. The idea in [57] is to approximate the local Sobolev regularity of the unknown analytical solution on each element in the mesh and then use it as a basis to decide whether to perform p -refinement or h -refinement. The decay rates of the Legendre series coefficients were used in [40, 57, 58] to determine the local smoothness. In Devine & Flaherty [31], a p -refinement-based spatial error estimate where the local error is taken to be the difference between two approximations of differing degrees was used.

6.4 Numerical Examples

In this section, the mesh & order adaptive WENO method is used to solve a number of numerical examples. The performance of the method is verified by comparing the results with those obtained from mesh-only adaptive methods. We compare the L_1 -, L_2 - and L_∞ -norm errors as well as the degrees of freedom. Using our computational experience in Chapter 3, we perform the WENO reconstruction for $\phi_1(r) = r$ with four stencils of size four. A constant is added to this interpolant resulting in five degrees of freedom (dof) in this case. Similarly, for reconstruction with $\phi_{2,2}(r) = r^2 \log(r)$ we utilise seven stencils of size seven and thus have ten degrees of freedom per stencil while for reconstruction with $\phi_{2,3}(r) = r^4 \log(r)$ we use nine stencils of size nine and as such we have 15 degrees of freedom per stencil. We use $c_1 = c_2 = 0.5$ in all our computations. The results in this chapter, especially for Burgers' equation, are preliminary.

	ϕ_1	$\phi_{2,2}$	$\phi_{2,3}$	dof	E_1	E_2	E_∞
ϕ_1 m-o	3106	—	—	15530	$4.9233 \cdot 10^{-3}$	$1.4188 \cdot 10^{-2}$	$3.2784 \cdot 10^{-1}$
$\phi_{2,2}$ m-o	—	1948	—	19480	$3.5932 \cdot 10^{-3}$	$8.9724 \cdot 10^{-3}$	$7.9720 \cdot 10^{-2}$
$\phi_{2,3}$ m-o	—	—	1786	26790	$2.2741 \cdot 10^{-3}$	$6.5481 \cdot 10^{-3}$	$5.8612 \cdot 10^{-2}$
m & o	1231	455	264	14665	$1.9057 \cdot 10^{-3}$	$5.6922 \cdot 10^{-3}$	$4.5337 \cdot 10^{-2}$

Table 6.1: Error comparison for mesh-only (m-o) and mesh & order (m & o) adaptivity for the linear advection equation. Two refinement cycles for each time step.

	ϕ_1	$\phi_{2,2}$	$\phi_{2,3}$	dof	E_1	E_2	E_∞
ϕ_1 m-o	3308	—	—	16540	$5.2775 \cdot 10^{-2}$	$1.7619 \cdot 10^{-2}$	$9.7894 \cdot 10^{-2}$
$\phi_{2,2}$ m-o	—	2026	—	20260	$3.3717 \cdot 10^{-3}$	$7.7156 \cdot 10^{-3}$	$7.3035 \cdot 10^{-2}$
$\phi_{2,3}$ m-o	—	—	1882	28230	$1.8462 \cdot 10^{-3}$	$5.8325 \cdot 10^{-3}$	$6.1822 \cdot 10^{-2}$
m & o	1405	488	283	16150	$1.8825 \cdot 10^{-3}$	$5.3236 \cdot 10^{-3}$	$4.7781 \cdot 10^{-2}$

Table 6.2: Error comparison for mesh-only (m-o) and mesh & order (m & o) adaptivity for the linear advection equation. Three refinement cycles for each time step.

6.4.1 Example 1: Linear advection

We solve the linear advection equation (3.35) with initial data

$$u_0(\mathbf{x}) = \begin{cases} \exp\left(\frac{\|\mathbf{x}-\mathbf{c}\|^2}{\|\mathbf{x}-\mathbf{c}\|^2-R^2}\right), & \|\mathbf{x}-\mathbf{c}\| < R; \\ 0, & \text{otherwise,} \end{cases} \quad (6.1)$$

with $R = 0.15$, $\mathbf{c} = (0.4, 0.4)^T$ on the computational domain $\Omega = [0, 1] \times [0, 1] \subset \mathbb{R}^2$. The simulation is run till time $t = 0.25$. The Gaussian shaped initial profile is advected along the diagonal of the computational domain without essentially changing shape. The centre of the base of the Gaussian-shaped function is initially at $(0.4, 0.4)$ and moves to $(0.65, 0.65)$ by the end of the simulation.

We solve this problem on a base mesh of 512 elements which we subject to five initial mesh refinement cycles. We compute the solution first of all with a WENO reconstruction using $\phi_1(r) = r$ before applying the mesh refinement and order variation. We ensure in all our computations that the order of neighbouring elements differs by at most one. In Tables 6.1 and 6.2, we demonstrate the performance of the mesh & order adaptive algorithm as compared to the mesh-only adaptive methods of Chapter 5 with two and three refinement cycles per time step.

We see a significant reduction in the degrees of freedom while providing errors similar to the highest order of reconstruction. In Figures 6.1 and 6.2, we present the mesh & order mesh showing the different orders of reconstruction for different cells generated at $t = 0.25$ after two and three refinement cycles respectively. The red, green and blue cells indicate cells where the reconstruction is performed with $\phi_1(r)$, $\phi_{2,2}(r)$ and $\phi_{2,3}(r)$ respectively. We observe that the reconstruction around the discontinuity is performed

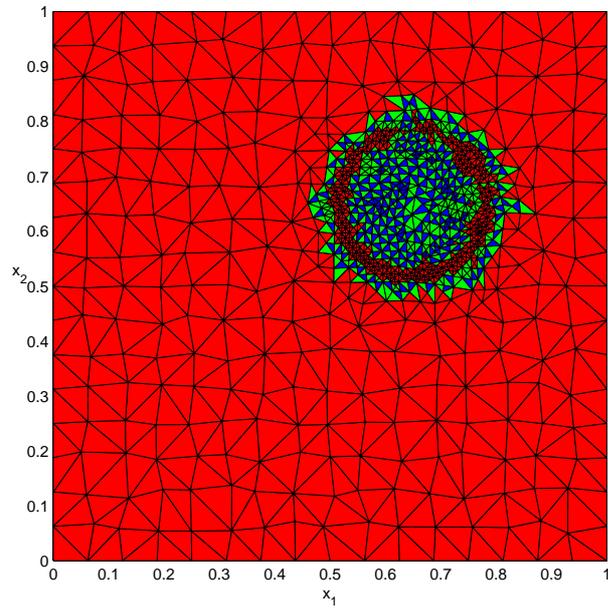


Figure 6.1: Mesh & order adaptivity mesh for linear advection problem. The red, green and blue cells are for reconstruction with $\phi_1(r)$, $\phi_{2,2}(r)$, $\phi_{2,3}(r)$ respectively.

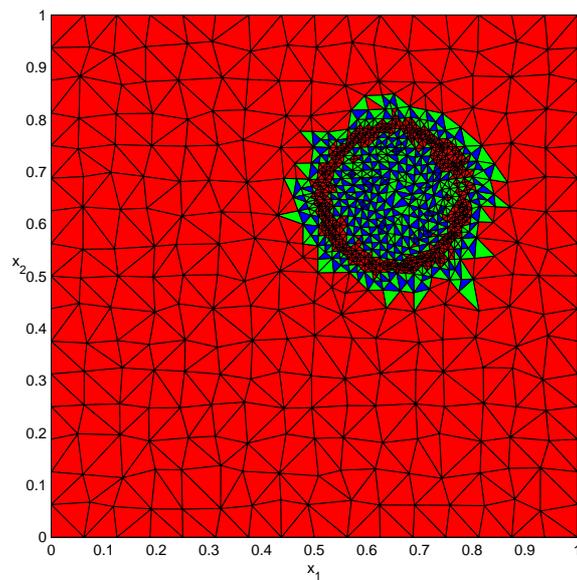


Figure 6.2: Mesh & Order adaptivity mesh for linear advection problem. The red, green and blue cells are for reconstruction with $\phi_1(r)$, $\phi_{2,2}(r)$, $\phi_{2,3}(r)$ respectively.

	ϕ_1	$\phi_{2,2}$	$\phi_{2,3}$	dof	E_1	E_2	E_∞
ϕ_1 m-o	2856	—	—	14280	$5.5145 \cdot 10^{-4}$	$2.6087 \cdot 10^{-3}$	$8.2692 \cdot 10^{-3}$
$\phi_{2,2}$ m-o	—	2294	—	22940	$4.0455 \cdot 10^{-4}$	$1.7554 \cdot 10^{-3}$	$6.0136 \cdot 10^{-3}$
$\phi_{2,3}$ m-o	—	—	2018	30270	$2.7332 \cdot 10^{-4}$	$1.0061 \cdot 10^{-3}$	$4.0947 \cdot 10^{-3}$
m & o	1470	528	204	15690	$2.5218 \cdot 10^{-4}$	$1.0237 \cdot 10^{-3}$	$3.9813 \cdot 10^{-3}$

Table 6.3: Error comparison for mesh-only (m-o) and mesh & order (m & o) adaptivity for the kinematic wave equation. Two refinement cycles for each time step.

	ϕ_1	$\phi_{2,2}$	$\phi_{2,3}$	dof	E_1	E_2	E_∞
ϕ_1 m-o	3082	—	—	15410	$4.5331 \cdot 10^{-4}$	$1.9881 \cdot 10^{-3}$	$7.4802 \cdot 10^{-3}$
$\phi_{2,2}$ m-o	—	2410	—	24100	$3.4011 \cdot 10^{-4}$	$8.3914 \cdot 10^{-4}$	$5.6018 \cdot 10^{-3}$
$\phi_{2,3}$ m-o	—	—	2216	33240	$1.9752 \cdot 10^{-4}$	$6.2371 \cdot 10^{-4}$	$3.8806 \cdot 10^{-3}$
mesh & order	1783	432	249	16970	$2.0418 \cdot 10^{-4}$	$6.9186 \cdot 10^{-4}$	$4.0344 \cdot 10^{-3}$

Table 6.4: Error comparison for mesh-only (m-o) and mesh & order (m & o) adaptivity for the kinematic wave equation. Three refinement cycles for each time step.

with very small low order elements while away from the discontinuity, larger higher order elements are used. Where the solution is zero (i.e. outside the support of the solution and its vicinity), there is no benefit using a high order reconstruction and so a low order recovery function is used.

6.4.2 Example 2: Kinematic wave equation

In this example, we consider the kinematic wave equation [16]

$$u_t + 2u_{x_1} + 2u_{x_2} = 0, \quad 0 < x_1, x_2 < 1, \quad t > 0, \quad (6.2)$$

with the initial and Dirichlet boundary conditions specified so that the exact solution is the very steep but smooth wave

$$u(t, \mathbf{x}) = \frac{1}{2}(1 - \tanh(100x_1 - 10x_2 - 180t + 5)), \quad (6.3)$$

inclined at an angle of approximately 84 degrees with the positive x_1 direction that moves from left to right at an angle of 45 degrees across the square domain as the time progresses. We solve this problem using both mesh adaptivity and mesh & order adaptivity for $0 < t \leq 0.10$.

Once again, we solve this problem on a base mesh of 512 elements which we subject to five initial mesh refinement cycles. In Tables 6.3 and 6.4, we compare the performance of the mesh & order adaptive algorithm and the mesh-only adaptive methods with two and three refinement cycles per time step. We once again see a significant reduction in the degrees of freedom while providing errors similar to the highest order of reconstruction.

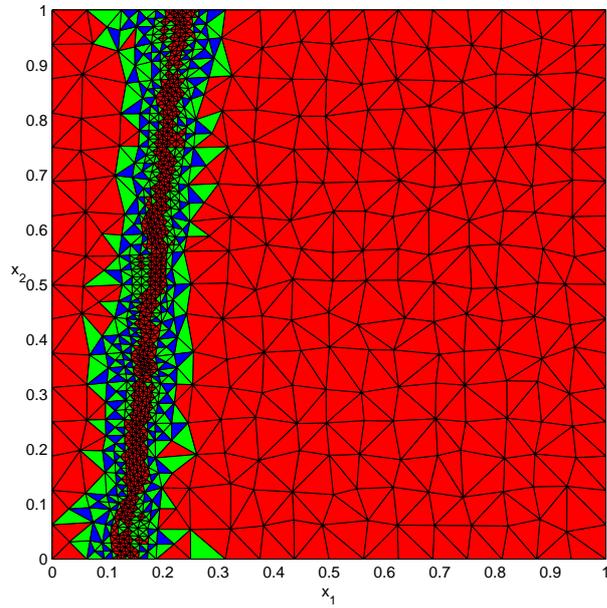


Figure 6.3: Mesh & Order adaptive mesh for kinematic wave equation. The red, green and blue cells are for reconstruction with $\phi_1(r)$, $\phi_{2,2}(r)$, $\phi_{2,3}(r)$ respectively.

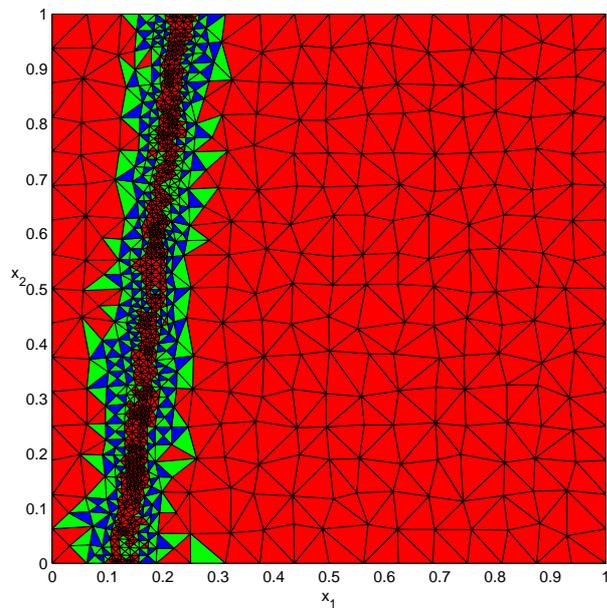


Figure 6.4: Mesh & Order adaptive mesh for kinematic wave equation. The red, green and blue cells are for reconstruction with $\phi_1(r)$, $\phi_{2,2}(r)$, $\phi_{2,3}(r)$ respectively.

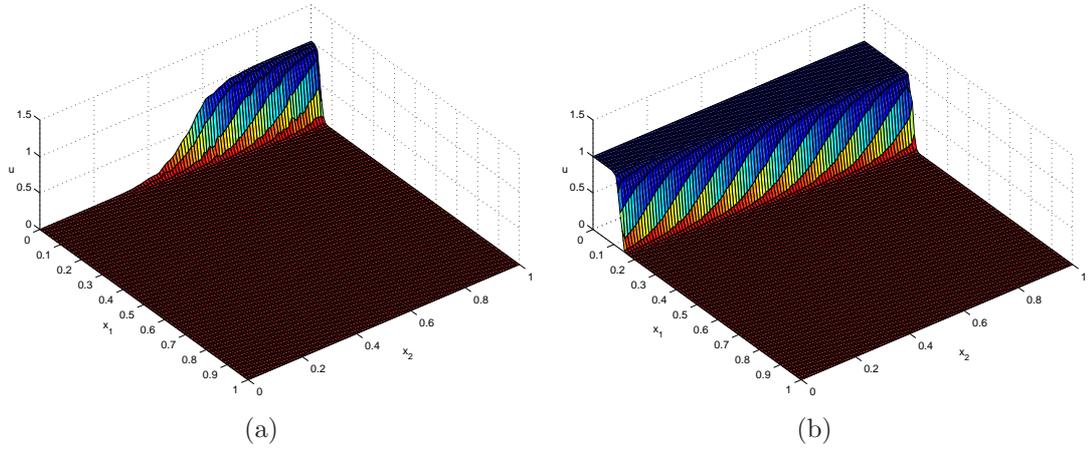


Figure 6.5: (a) Initial condition and (b) solution at $t = 0.1$ of the kinematic wave equation.

	ϕ_1	$\phi_{2,2}$	$\phi_{2,3}$	dof	E_1	E_2	E_∞
ϕ_1 m-o	2436	—	—	12180	$5.2811 \cdot 10^{-3}$	$3.9832 \cdot 10^{-2}$	$8.1573 \cdot 10^{-2}$
$\phi_{2,2}$ m-o	—	1848	—	18480	$3.0155 \cdot 10^{-3}$	$2.4811 \cdot 10^{-2}$	$7.5725 \cdot 10^{-2}$
$\phi_{2,3}$ m-o	—	—	1792	26880	$2.3709 \cdot 10^{-3}$	$1.3494 \cdot 10^{-2}$	$6.3697 \cdot 10^{-2}$
m & o	1135	467	218	13615	$2.7682 \cdot 10^{-3}$	$1.5214 \cdot 10^{-2}$	$6.5002 \cdot 10^{-2}$

Table 6.5: Error comparison for mesh-only (m-o) and mesh & order (m & o) adaptivity for the Burgers' equation. Two refinement cycles for each time step.

In Figures 6.3 and 6.4, we present the mesh & order mesh showing the different orders of reconstruction for different cells generated at $t = 0.1$ for two and three refinement cycles respectively. The initial condition and the solution at time $t = 0.1$ are shown in Figure 6.5. We see that the steep wave is well resolved and there are also no unwanted oscillations.

6.4.3 Example 3: Burgers' equation

Finally, we consider the Burgers' equation (3.37) with initial condition (5.3) on the computational domain $\Omega = [-0.5, 0.5] \times [-0.5, 0.5] \subset \mathbb{R}^2$ and on the time interval $I = [0, 0.6]$.

In Figures 6.6 and 6.7, we present the mesh & order mesh showing the different orders

	ϕ_1	$\phi_{2,2}$	$\phi_{2,3}$	dof	E_1	E_2	E_∞
ϕ_1 m-o	2580	—	—	12900	$4.8246 \cdot 10^{-3}$	$3.7335 \cdot 10^{-2}$	$7.6023 \cdot 10^{-2}$
$\phi_{2,2}$ m-o	—	1910	—	19100	$2.5049 \cdot 10^{-3}$	$2.1843 \cdot 10^{-2}$	$6.5741 \cdot 10^{-2}$
$\phi_{2,3}$ m-o	—	—	1856	27840	$1.8667 \cdot 10^{-3}$	$1.2694 \cdot 10^{-2}$	$5.8862 \cdot 10^{-2}$
mesh & order	1628	474	206	15970	$1.9653 \cdot 10^{-3}$	$1.4175 \cdot 10^{-2}$	$6.1239 \cdot 10^{-2}$

Table 6.6: Error comparison for mesh-only (m-o) and mesh & order (m & o) adaptivity for the Burgers' equation. Three refinement cycles for each time step.

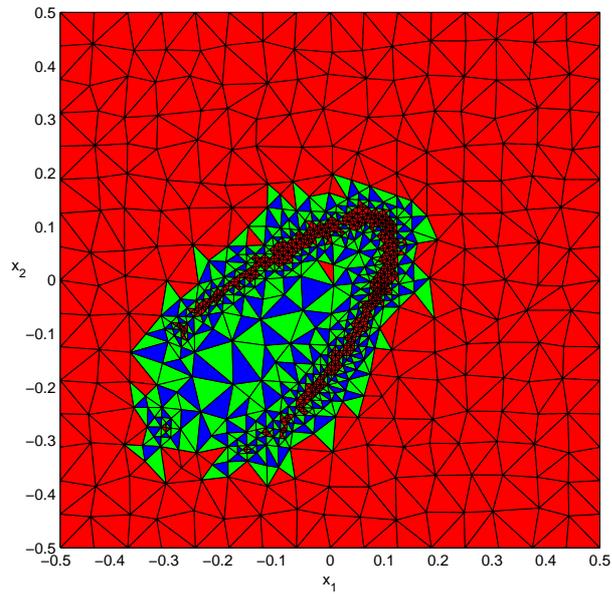


Figure 6.6: Mesh & Order adaptive mesh for Burgers' equation. The red, green and blue cells are for reconstruction with $\phi_1(r)$, $\phi_{2,2}(r)$, $\phi_{2,3}(r)$ respectively.

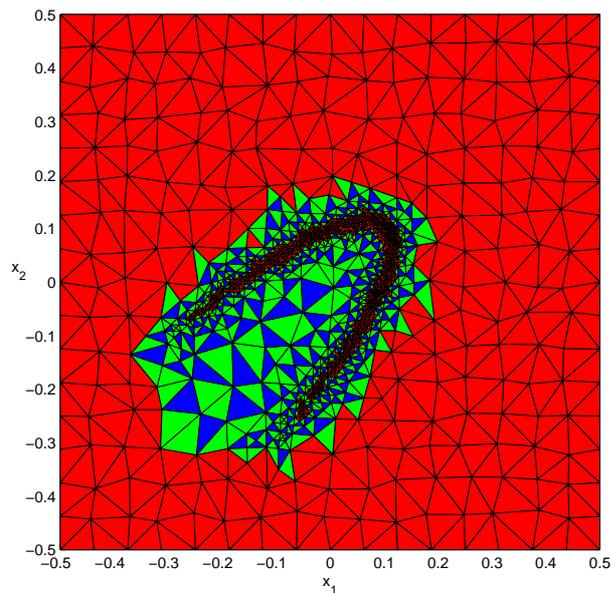


Figure 6.7: Mesh & Order adaptive mesh for Burgers' equation. The red, green and blue cells are for reconstruction with $\phi_1(r)$, $\phi_{2,2}(r)$, $\phi_{2,3}(r)$ respectively.

of reconstruction for different cells generated at $t = 0.6$ for two and three refinement cycles respectively. We see again from both figures that the reconstruction around the discontinuity is performed with very small low order elements while away from the discontinuity but either near or within the support of the solution, larger higher order elements are used. Moreover, outside the support of the function, we perform the reconstruction with $\phi_1(r)$ as we derive no benefit from using high order elements. The numerical results are shown in Tables 6.5 and 6.6.

Chapter 7

Conclusion and Outlook

7.1 Conclusion

The main goal of this thesis has been the coupling of the power of radial basis function interpolation with non-oscillatory finite volume methods for the numerical solution of the hyperbolic conservation laws. Radial basis functions are known to be suitable for reconstruction on both structured and unstructured grids and can also be effectively implemented on complex computational domains. We have also shown in this work that adaptivity can be a very useful attribute of a numerical scheme for hyperbolic conservation laws. Since the solutions of conservation laws may develop discontinuities in finite time, adaptive methods can effectively capture these critical features with high resolution and accuracy.

In Chapter 2, we introduced the concept of interpolation with radial basis functions. We provided the necessary theoretical background and looked at the conditions for the existence of the solution of the interpolation problem. We proceeded to concentrate on generalized interpolation where the interpolatory data does not consist of point evaluations but more general observation functionals. For the purposes of this work, we were specifically interested in the case where the functionals are cell average operators.

The finite volume methods in this thesis are based on local reconstruction methods using polyharmonic splines. To this end, we restated existing results concerning local Lagrange interpolation by polyharmonic splines in terms of cell average operators instead of point evaluations. We thereafter proved new results concerning the Lagrange basis functions, approximation order and numerical stability of derivatives of the polyharmonic spline interpolant. These results are essential in the implementation of the numerical methods in Chapter 4.

Chapter 3 was a reformulation of the RK-WENO method where the reconstruction step of the spatial discretisation was done using polyharmonic splines rather than polynomials. The implementation of this reconstruction method relies on the results

in Chapter 2 on the stable evaluation of polyharmonic splines. We provided a detailed treatment of all issues relating to the RK-WENO method such as time stepping and efficient stencil selection strategies. The polyharmonic spline reconstruction was found to be useful because of its flexibility, its optimality properties and because its native space provides a natural choice for the oscillation indicator of the WENO reconstruction. Our implementation of the RK-WENO method yielded the expected order of convergence for standard test problems. To further test the robustness of the RK-WENO method, we used it to solve Doswell's frontogenesis problem which exhibits a multiscale behaviour.

We further sought to combine our stable and essentially non-oscillatory polyharmonic spline reconstruction with a different time stepping and flux evaluation strategy. To this end, in Chapter 4, we combine the ADER time discretisation with the polyharmonic spline WENO reconstruction method. First, we expressed the numerical solution at all the spatial Gaussian integration points as a truncated Taylor series in time. The time derivatives were then replaced by space derivatives by the successive differentiation of the governing equation using the Cauchy-Kowalewski procedure. The approximate value of the solution and its space derivatives at cell interfaces are obtained by solving a set of Generalized Riemann Problems where the polyharmonic spline WENO interpolant and its derivatives are used for initial data. The solution of the GRPs forms the basis of the high order flux evaluation of the ADER-WENO method. We obtained second order convergence for reconstruction with $\phi_{2,2}(r) = r^2 \log(r)$ and third order convergence for $\phi_{2,3}(r) = r^4 \log(r)$ when the method was used to solve the linear advection equation and Burgers' equation. We also showed the good performance of the method when used in the numerical simulation of Smolarkiewicz's deformational flow test.

In Chapters 5 and 6, we demonstrated the utility of adaptivity as a tool for improving the quality of numerical schemes for conservation laws. We examined three types of adaptivity.

1. *Stencil adaptivity*: Here we used the flexibility in the choice of stencil sizes for polyharmonic spline reconstruction to use larger stencils for reconstruction in regions where the solution has steep gradients or discontinuities and used smaller stencils in regions of the computational domain where the solution is smooth. We showed that this was an effective way of reducing computational cost without reducing the quality of our numerical computations.
2. *Mesh adaptivity*: Using a suitable error indicator, the quality of the numerical approximation using the RK-WENO and ADER-WENO methods is enhanced and the computational costs reduced by refining and coarsening the computational mesh over some portions of the computational domain on the basis of an appropriate criterion. The performance of mesh adaptivity was validated with several

numerical examples. We also looked at the application of adaptive methods in the simulation of two-phase flow through porous media. Further enhancements were provided by coupling mesh adaptivity and stencil adaptivity.

3. *Mesh & Order adaptivity*: We presented some results on the design and implementation of an RK-WENO method that combined mesh adaptivity and order variation. The idea of mesh & order adaptivity was motivated by *hp*-adaptivity in finite element methods. We simultaneously adjusted the local grid resolution and the order of the recovery functions on the basis of suitable error estimates. In the end, we were able to achieve good accuracy along with a reduction in the degrees of freedom. To the best of our knowledge, this is the first instance of mesh & order adaptivity for finite volume methods.

7.2 Outlook

This thesis has focused mainly on the use of polyharmonic splines in the WENO reconstruction method. We have shown the suitability of using this class of RBFs in the development of finite volume methods. One possible area of further study is to use other radial basis functions in the WENO reconstruction method. We note however that one of the main reasons we worked with polyharmonic splines is the explicit knowledge of its native space which also provides a natural choice for the oscillation indicator. In most other cases of interest, the native spaces of the conditionally positive definite functions are usually characterizable via properties of the Fourier transform. This characterization is therefore quite implicit and does not allow for functions on bounded domains explicitly. Therefore, the suitable choice of oscillation indicator will need to be investigated for any other radial recovery function one may wish to use.

The computation of optimal weights for polyharmonic spline WENO reconstruction along the lines of Hu & Shu [59] needs to be investigated.

As concerning the ADER-WENO method, further investigation is needed on the appropriate oscillation indicators and weights for the derivatives of the polyharmonic spline WENO reconstruction. This may possibly improve the overall quality of the ADER-WENO method. At the moment we use the same oscillation indicator and weights for both the WENO reconstruction and all its weights. It will also be desirable to extend the ADER-WENO method to higher order polyharmonic splines ($k \geq 4$). Successful implementation of the ADER-WENO method with higher order polyharmonic splines will make it a contender for very high order time stepping.

There is also a need to extend the numerical methods in this thesis to conservation laws in higher space dimensions especially \mathbb{R}^3 where there may be several useful applica-

tions. Extension to nonlinear hyperbolic systems as well as problems with source terms may be a useful research direction to pursue.

One area of further research could be the coupling the mesh adaptivity with time adaptivity in order to use small time steps only for small cells and allow for larger time steps in larger cells in order to reduce numerical diffusion. In the implementation of mesh adaptivity in Chapter 5, the smallest cell was used in determining the time step and this yielded an increase in the simulation time.

We have shown that the finite volume methods with RBF reconstruction can be implemented with several adaptive techniques. It will therefore be useful to see how mesh movement (r -refinement) can be combined with our RBF based finite volume methods.

Appendix A

Notations and Abbreviations

A.1 Notations

$[L^\infty(\mathbb{R}^d)]$	Space of measurable functions on \mathbb{R}^d	3
$[L^\infty(\mathbb{R}^+ \times \mathbb{R}^d)]$	Space of measurable functions on \mathbb{R}^d and \mathbb{R}^+	3
$[C_0^\infty([0, \infty) \times \mathbb{R}^d)]$	Space of smooth functions on \mathbb{R}^d and \mathbb{R}^+ with compact support .	3
$[C^1(\mathbb{R})]$	Space of continuous functions on \mathbb{R}	3
$[L^1(\mathbb{R}^d)]$	Space of integrable functions on \mathbb{R}^d	4
$[BV(\mathbb{R}^d)]$	Space of functions with bounded variation on \mathbb{R}^d	4
$[\mathcal{P}_m^d]$	Space of all real-valued polynomials in d variables of degree at most $m - 1$...	17
$[C^\infty]$	Space of smooth functions	20
$[\mathcal{N}_\phi]$	Native Hilbert space for conditionally positive definite function ϕ	22
$[\phi_{d,k}]$	Polyharmonic spline in dimension d and of order k	23
$[\text{BL}_k(\mathbb{R}^d)]$	Beppo Levi space of order k on \mathbb{R}^d	24
$[\text{BL}_2(\mathbb{R}^2)]$	Beppo Levi space of order 2 on \mathbb{R}^2	24
$[\mathcal{P}_k^d]$	Space of all real-valued polynomials in d variables of degree at most $k - 1$	31

A.2 Abbreviations

[WENO] Weighted Essentially Non-Oscillatory Method	1
[ADER] Arbitrary high order using high order DERivatives	2
[FEM] Finite Element Method	7
[FVM] Finite Volume Method	7
[PDE] Partial Differential Equation	7
[PPM] Piecewise Parabolic Method	10
[ENO] Essentially Non-Oscillatory Method	10
[RBF] Radial Basis Functions	17
[SSPRK] Strong Stability Preserving Runge-Kutta method	51
[TVD] Total Variation Diminishing	51
[CFL] Courant-Friedrich-Lewy condition	51
[GRP] Generalized Riemann Problem	79

Appendix B

Newest Vertex Bisection

In this appendix, we briefly describe the mesh adaptation strategy used in Chapter 6 for mesh & order adaptivity. The method, known as *newest vertex bisection*, was first proposed by Sewell [106] in 1972. However, the description of the newest vertex bisection method presented in this appendix is based on the work of Chen & Zhang [20, 21, 22]. In this series of papers, they provide a MATLAB implementation of the newest vertex bisection for the refinement and coarsening of triangular meshes.

Suppose we have a polygonal domain $\Omega \subset \mathbb{R}^2$ with a conforming triangulation \mathcal{T} . We denote the set of vertices of the triangulation by $\mathcal{V}(\mathcal{T})$ and the set of all edges by $\mathcal{E}(\mathcal{T})$. For a vertex \mathbf{x} and an edge $\mathbf{e} \in \mathcal{E}(\mathcal{T})$, we define the first rings of \mathbf{x} and \mathbf{e} as

$$\mathcal{R}_{\mathbf{x}} = \{T \in \mathcal{T} \mid \mathbf{x} \in T\}, \quad \mathcal{R}_{\mathbf{e}} = \{T \in \mathcal{T} \mid \mathbf{e} \in T\},$$

and we define the local patches for \mathbf{x} and \mathbf{e} as

$$\omega_{\mathbf{x}} = \bigcup_{T \in \mathcal{R}_{\mathbf{x}}} T, \quad \omega_{\mathbf{e}} = \bigcup_{T \in \mathcal{R}_{\mathbf{e}}} T.$$

We observe that $\omega_{\mathbf{x}}$ and $\omega_{\mathbf{e}}$ are subdomains of Ω while $\mathcal{R}_{\mathbf{x}}$ and $\mathcal{R}_{\mathbf{e}}$ are the set of triangles. For each $\mathbf{x} \in \mathcal{V}(\mathcal{T})$, the valence of \mathbf{x} is defined as the number of elements in $\mathcal{R}_{\mathbf{x}}$, i.e. $\#\mathcal{R}_{\mathbf{x}}$.

B.1 Bisection of Triangles

The refinement of marked triangles of a triangular mesh (on the basis of an error indicator) by the newest vertex bisection starts with the *labeling* of the triangulation \mathcal{T} . To this end, for each element $T \in \mathcal{T}$, we label one vertex of T as the *newest vertex* and we

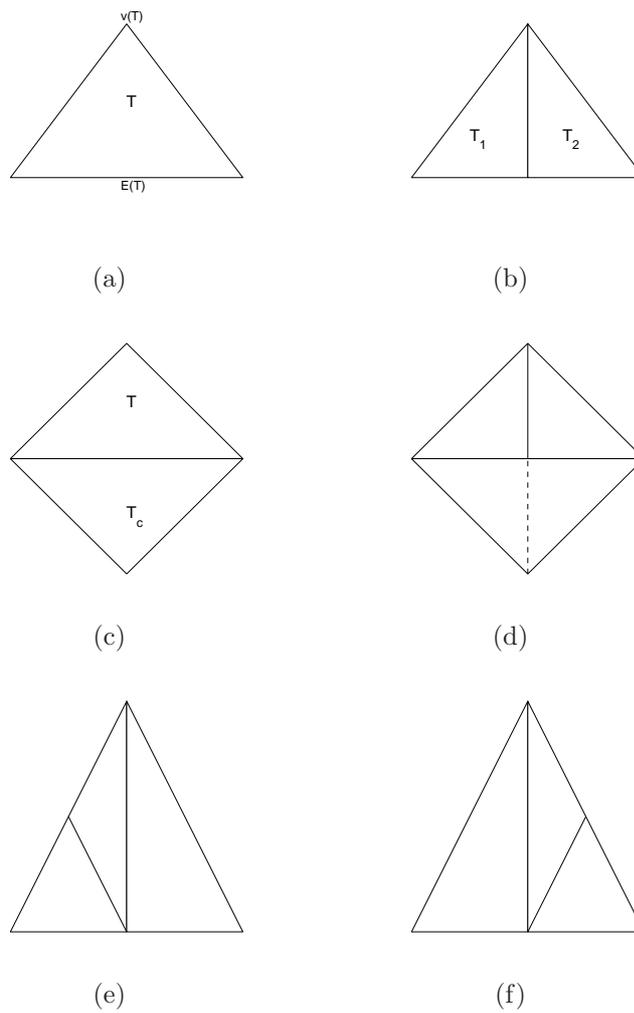


Figure B.1: Newest vertex bisection. (a) Labeled triangle T , (b) children of T , (c) marking for conformity, (d) bisection for conformity, (e) left child bisection and (f) right child bisection.

call it $\mathbf{v}(T)$ while the opposite edge of $\mathbf{v}(T)$ is called the *refinement edge* and denoted $\mathbf{E}(T)$, see Figure B.1(a). Once this labeling is completed, the refinement of the marked triangles follows the next two steps:

- A triangle (the *father*) is bisected to generate two elements (the *children*) by connecting its newest vertex with the midpoint of its refinement edge. Children with the same father are called *brothers* to each other. T_1 and T_2 in Figure B.1(b), are the children of T in Figure B.1(a).
- The new vertex created at the midpoint of the refinement edge is then labeled as the newest vertex of each child. Thus, they can be bisected as shown in Figure B.1(e) and Figure B.1(f).

Given an initial labeled triangulation \mathcal{T}_0 , we can define

$$\mathfrak{F}(\mathcal{T}_0) := \{\mathcal{T} \mid \mathcal{T} \text{ is obtained from } \mathcal{T}_0 \text{ by the newest vertex bisection}\}.$$

$\mathfrak{F}(\mathcal{T}_0)$ is not necessarily conforming. In order ensure conformity, which is necessary for both refinement and coarsening, we introduce the concept of *compatible bisection*.

Given a labeled initial mesh \mathcal{T}_0 , we define

$$\mathfrak{T}(\mathcal{T}_0) := \{\mathcal{T} \in \mathfrak{F}(\mathcal{T}_0) \mid \mathcal{T} \text{ is conforming}\}.$$

Let \mathcal{T} be a labeled conforming triangulation. Then, for any $T \in \mathcal{T}$, we define a map

$$F(T) = \begin{cases} T', & \mathbf{E}(T) \subset T' \in \mathcal{T}; \\ \emptyset, & \mathbf{E}(T) \in \partial\Omega. \end{cases}$$

This means that $T' \in \mathcal{T}$ is an element (if it exists) which has the refinement edge of T as its own edge. Although, $\mathbf{E}(T) \subset T'$, the refinement edge of T' could be different from $\mathbf{E}(T)$, the refinement edge of T .

An element T is said to be *compatible* if $F(T) = \emptyset$ or $F(F(T)) = T$, i.e. $\mathbf{E}(T) = \mathbf{E}(T')$. The refinement edge $\mathbf{E}(T)$ of a compatible element is called a compatible edge and $\omega_{\mathbf{E}}$ is called a compatible patch. Thus, the first ring $\mathcal{R}_{\mathbf{E}}$ is a pair of triangles sharing the same refinement edge or a triangle whose refinement edge is on the boundary. Bisecting every element in the first ring $\mathcal{R}_{\mathbf{E}}$ gives a new conforming triangulation. This is what is known as a compatible bisection. For example, in Figures B.1(c) T and T_c share the same refinement edge and B.1(d) shows their compatible bisection.

A compatible bisection can therefore be defined as a map

$$b : \mathcal{R}_{\mathbf{E}} \mapsto \mathcal{R}_{\bar{\mathbf{x}}}$$

where $\tilde{\mathbf{x}}$ is the midpoint of the refinement (i.e. the new node).

B.2 Coarsening Algorithm

In [22], coarsening can only be done by the removal of *good-for-coarsening points* or *good points* for short, that is, the removal of these points will not destroy the conformity of the triangulation.

Definition B.1 (Good-for-coarsening nodes) *For a triangulation $\mathcal{T} \in \mathfrak{T}(\mathcal{T}_0)$, a node $\mathbf{x} \in \mathcal{V}(\mathcal{T})$ is called a good-for-coarsening node or a good node, if there exists a compatible bisection b and compatible patch \mathcal{R}_e such that $\mathcal{R}_x = b(\mathcal{R}_e)$. The set of all good nodes in the grid \mathcal{T} will be denoted by $G(\mathcal{T})$.*

The following theorem of Chen & Zhang [21, 22] provides a characterization of good nodes.

Theorem B.2 *Let \mathcal{T}_0 be a compatible labeled conforming triangulation. For any $\mathcal{T} \in \mathfrak{T}(\mathcal{T}_0)$ and $\mathcal{T} \neq \mathcal{T}_0$, the set of nodes $G(\mathcal{T})$ is not empty. Furthermore, $\mathbf{x} \in G(\mathcal{T})$ if and only if*

1. *it is not a node of the initial triangulation \mathcal{T}_0 ;*
2. *it is the newest vertex of all elements in \mathcal{R}_x ;*
3. *its valence is 4 (for an interior node) or 2 (for a boundary node).*

After all the good nodes have been obtained, we traverse all the triangles in the triangulation and find those that are marked for coarsening and also contain good points. If T is one such element, we need to find its brother T' . Thus, when the good node is removed, the father of T and T' is restored. In other words, if we define the set

$$G_c(\mathcal{T}) = \{\mathbf{x} \in G \mid \mathbf{x} \text{ is a vertex of a triangle marked for coarsening}\},$$

then for $\mathbf{x} \in G_c(\mathcal{T})$, we perform the coarsening by replacing \mathcal{R}_x by $b^{-1}(\mathcal{R}_x)$ where b is a compatible bisection.

It was also proved in [22] that the initial grid can be obtained by using the coarsening algorithm recursively and the grid obtained from this type of coarsening is conforming.

Appendix C

Modelling Two-Phase Flow in Porous Media

C.1 Introduction

The mathematical modelling of two-phase flow has applications in several branches of science and engineering. We are particularly interested in modelling an enhanced oil recovery process where a wetting fluid, say water, is injected into an oil-saturated porous medium in order to displace the oil (the non-wetting fluid) so that it can be collected. By a porous medium, we mean a solid with many small connected pores through which fluid may flow. We will assume that there is no mass transfer between the fluids although this could occur if some kind of chemical reaction takes place between them.

The flow of two immiscible fluid phases in a porous medium is modelled by the equations of conservation of mass and the generalized Darcy's law. In our discussion, we denote the wetting phase (water) by w and the non-wetting phase (oil) by n .

C.2 The Governing Equations

If we take a control volume in the porous media, with porosity $\bar{\phi}(\mathbf{x})$, the equation of mass conservation of water is

$$\frac{\partial}{\partial t} \bar{\phi}(\mathbf{x}) u_w(t, \mathbf{x}) + \nabla \cdot \mathbf{v}_w(t, \mathbf{x}) = S_w, \quad (\text{C.1})$$

while the equation of mass conservation of oil is

$$\frac{\partial}{\partial t} \bar{\phi}(\mathbf{x}) u_n(t, \mathbf{x}) + \nabla \cdot \mathbf{v}_n(t, \mathbf{x}) = S_n. \quad (\text{C.2})$$

For any homogeneous medium the porosity $\bar{\phi}$ is a constant. The terms S_w and S_n represent the sources or sinks which we assume to be zero. In (C.1) and (C.2), $\mathbf{v}_w(t, \mathbf{x})$ and $\mathbf{v}_n(t, \mathbf{x})$ are the velocities of the wetting and non-wetting phases and $u_w(t, \mathbf{x})$ and $u_n(t, \mathbf{x})$ are the saturations of the two phases. The saturation of a phase is the fraction of the pores of the porous medium filled by that phase. If we assume the void space in the medium is filled by only those two fluids, we have

$$u_w(t, \mathbf{x}) + u_n(t, \mathbf{x}) = 1. \quad (\text{C.3})$$

The pressure of the non-wetting fluid is usually greater than the pressure of the wetting fluid and the difference between the two pressures is known as the capillary pressure,

$$p_c = p_n - p_w, \quad (\text{C.4})$$

which is a unique function of the saturation [93].

The phase velocities $\mathbf{v}_w(t, \mathbf{x})$ and $\mathbf{v}_n(t, \mathbf{x})$ are modelled using Darcy's law. For a single-phase flow, Darcy's law relates the phase velocity \mathbf{v} to the pressure p , the viscosity μ , the density ρ , the permeability K and the depth of the fluid D via

$$\mathbf{v} = -\frac{K}{\mu}(\nabla p - \rho g \nabla D), \quad (\text{C.5})$$

where g is the gravitational constant. Darcy's law can then be extended to two-phase flow as follows:

$$\mathbf{v}_w = -\frac{K_w}{\mu_w}(\nabla p_w - \rho_w g \nabla D), \quad (\text{C.6})$$

$$\mathbf{v}_n = -\frac{K_n}{\mu_n}(\nabla p_n - \rho_n g \nabla D), \quad (\text{C.7})$$

where K_w and K_n are the effective permeabilities of the two phases. These must be less than or equal to the single fluid permeability K since the concurrent flow of two fluids causes each to affect the flow of the other. If \mathbf{K} is the permeability of the porous medium, then the relative permeabilities of each phase are defined as

$$k_w = \frac{K_w}{\mathbf{K}} \leq 1, \quad (\text{C.8})$$

$$k_n = \frac{K_n}{\mathbf{K}} \leq 1, \quad (\text{C.9})$$

and are regarded as functions of the saturations $u_w(t, \mathbf{x})$ and $u_n(t, \mathbf{x})$ respectively. We can use the relative permeabilities and the explicit dependence of the quantities on t, \mathbf{x} ,

u_w and u_n to re-write (C.6) as

$$\begin{aligned}\mathbf{v}_w(t, \mathbf{x}) &= -\mathbf{K}(\mathbf{x}) \frac{k_w(u_w)}{\mu_w} (\nabla p_w(t, \mathbf{x}) - \rho_w(t, \mathbf{x}) g \nabla D), \\ \mathbf{v}_n(t, \mathbf{x}) &= -\mathbf{K}(\mathbf{x}) \frac{k_n(u_n)}{\mu_n} (\nabla p_n(t, \mathbf{x}) - \rho_n(t, \mathbf{x}) g \nabla D),\end{aligned}\tag{C.10}$$

where μ_w and μ_n are the phase viscosities. Furthermore, we define the phase mobilities, M_w and M_n , which will be functions of the saturations, by the ratios

$$M_w(u_w) = \frac{k_w(u_w)}{\mu_w} \quad \text{and} \quad M_n(u_n) = \frac{k_n(u_n)}{\mu_n}.$$

and the total mobility M_t is defined as $M_t = M_w + M_n$. We can also define an average pressure \mathbf{p} by

$$\mathbf{p} = \frac{p_w + p_n}{2}.$$

To this end, the pressure of the two fluids can be expressed in terms of the average pressure and capillary pressure as

$$\begin{aligned}p_w &= \mathbf{p} - \frac{1}{2} p_c, \\ p_n &= \mathbf{p} + \frac{1}{2} p_c.\end{aligned}$$

C.3 The Simplified Mathematical Model

We can simplify the model by assuming that:

1. the porosity $\bar{\phi}$ is constant throughout the medium i.e. the medium is homogeneous,
2. the capillary effects can be ignored ($p_c = 0$),
3. the gravitational effects can be ignored ($g = 0$).

Therefore, if $\mathbf{v}_w + \mathbf{v}_n = \mathbf{v}$, then using (C.1), (C.2) and (C.3) we obtain the *incompressibility relation*

$$\nabla \cdot \mathbf{v}(t, \mathbf{x}) = 0,\tag{C.11}$$

which means that the total fluid velocity $\mathbf{v}(t, \mathbf{x})$ is divergence-free. Using *Darcy's law* (C.10), we obtain

$$\mathbf{v} = -\mathbf{K}(\mathbf{x}) M_t(u) \nabla \mathbf{p}(t, \mathbf{x}).\tag{C.12}$$

Finally, the velocity of the wetting phase is given as

$$\mathbf{v}_w(t, \mathbf{x}) = \mathbf{v}(t, \mathbf{x}) \cdot F_w(u_w),\tag{C.13}$$

where $F_w(u_w)$ is the flux and is given as

$$F_w(u_w) = \frac{M_w}{M_t}.$$

For the sake of notational simplicity, we take u to be the saturation of the wetting phase, i.e. $u \equiv u_w$, and so $u_n = 1 - u$. Furthermore, we let $F(u) = F_w(u_w)$. Generally the form of k and F are determined by experiment. However, for two-phase flows, the relative permeabilities k_w and k_n given by the quadratic functions

$$k_w = u^2 \quad \text{and} \quad k_n = (1 - u)^2$$

offer a reliable and widely used model [93]. This implies that

$$M_t(u) = \frac{u^2}{\mu_w} + \frac{(1 - u)^2}{\mu_n} \quad (\text{C.14})$$

and the fractional flow function F in (C.13) is given by

$$F(u) = \frac{u^2}{u^2 + m(1 - u)^2}$$

where $m = \mu_w/\mu_n$ is the ratio of the viscosities of the two fluids. Suppose we assume $\bar{\phi} \equiv 1$, then from (C.1) we obtain

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla F(u) = 0. \quad (\text{C.15})$$

The saturation equation (C.15) is referred to as the Buckley-Leverett equation [17].

Thus, assuming the absence of capillary and gravitational effects, (C.11), (C.12) and (C.15) model the flow of two fluids, water and oil, through a porous homogeneous medium with $\bar{\phi} = 1$.

Bibliography

- [1] R. Abgrall. On ENO schemes on unstructured meshes: analysis and implementation. *J. Comp. Phys.*, 114:45–58, 1994.
- [2] N. Adams and K. Shariff. A high resolution hybrid compact-ENO scheme for shock turbulence interaction problems. *J. Comp. Phys*, 127:27–51, 1996.
- [3] R. Adams. *Sobolev Spaces*. Academic Press, New York, 1975.
- [4] S. Adjerid, J. E. Flaherty, P. K. Moore, and Y. J. Wang. High-order adaptive methods for parabolic systems. *Physica D*, 60:94–111, 1992.
- [5] M. Ainsworth and W. Senior. An adaptive refinement strategy for *hp* finite element computations. *Appl. Numer. Math.*, 26:165–178, 1997.
- [6] R. Artebrant and H. J. Schroll. Limiter-free third order logarithmic reconstruction. *SIAM J. Sci. Comput.*, 28:359–381, 2006.
- [7] M. J. Baines, S. J. Leary, and M. E. Hubbard. Multidimensional least square fluctuation distribution schemes with adaptive mesh movement for steady hyperbolic problems. *SIAM J. Sci. Comput.*, 23:1485–1502, 2002.
- [8] R. E. Bank and A. Weiser. Some *a posteriori* error estimators for elliptic partial differential equations. *Math. Comp.*, 44:283–301, 1985.
- [9] T. J. Barth. An introduction to upwind finite volume and finite element methods: some unifying and contrasting themes. In *VKI Lectures Series on High Order Methods for Computational Physics*, 2005.
- [10] T. J. Barth and H. Deconinck. *High Order Methods for Computational Physics*. Springer–Verlag, 1999.
- [11] R. Beatson and M. K. Langton. Integral interpolation. In A. Iske and J. Levesley, editors, *Algorithms for Approximation V*, pages 199–218. Springer-Verlag, Heidelberg, 2006.

-
- [12] J. Behrens and A. Iske. Grid-free adaptive semi-Lagrangian advection using radial basis functions. *Comp. Math. Appl.*, 43:319–327, 2002.
- [13] M. Ben-Artzi and J. Falcovitz. A second-order Godunov-type scheme for compressible fluid dynamics. *J. Comp. Phys.*, 55:1–32, 1984.
- [14] K. Bey. *An hp-adaptive discontinuous Galerkin method for hyperbolic conservation laws*. PhD thesis, University of Texas, Austin, 1994.
- [15] F. Bianco, G. Puppo, and G. Russo. High order central schemes for hyperbolic system of conservation laws. *SIAM J. Sci. Comput.*, 21:294–322, 1999.
- [16] R. Biswas, K.D. Devine, and J.E. Flaherty. Parallel, adaptive finite element methods for conservation laws. *Appl. Numer. Math.*, 14:255–283, 1994.
- [17] M. D. Buckley and M. C. Leverett. Mechanisms of fluid displacements in sands. *Trans. AIME*, 146:107–116, 1942.
- [18] M. D. Buhmann. A new class of radial basis functions with compact support. *Math. Comp.*, 70:307–318, 2000.
- [19] M. D. Buhmann. *Radial Basis Functions*. Cambridge University Press, 2003.
- [20] L. Chen. Short bisection implementation in MATLAB. Research notes, 2006.
- [21] L. Chen and C. S. Zhang. AFEM@MATLAB: a MATLAB package of adaptive finite element methods. Technical report, University of Maryland, 2006.
- [22] L. Chen and C. S. Zhang. A coarsening algorithm and multilevel methods on adaptive grids by newest vertex bisection. Preprint, 2007.
- [23] B. Cockburn, S. Hou, and C. W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. *Math. Comp.*, 54:545–581, 1990.
- [24] B. Cockburn, J. Jerome, and C. W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One dimensional systems. *J. Comp. Phys.*, 84:90–113, 1989.
- [25] B. Cockburn, G. E. Karniadakis, and C. W. Shu. *Discontinuous Galerkin Methods: Theory, Computation and Applications*. Springer-Verlag, 2000.
- [26] A. Cohen, S. M. Kaber, S. Müller, and M. Postel. Fully adaptive multiresolution finite volume schemes for conservation laws. *Math. Comp.*, 72:183 – 225, 2003.

- [27] P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *J. Comp. Phys*, 87:171–200, 1990.
- [28] P. Colella and P. Woodward. The piecewise parabolic method for gas-dynamical simulations. *J. Comp. Phys*, 54:174–201, 1984.
- [29] M. Crandall and A. Majda. Monotone difference approximations for scalar conservation laws. *Math. Comp.*, 34:1–22, 1980.
- [30] A. Dedner and M. Ohlberger. A new *hp*-adaptive DG scheme for conservation laws based on error control. In *Proc. of the Eleventh International Conference on Hyperbolic Problems: Theory, Numerics, Applications*, pages 187–198, 2006.
- [31] K. D. Devine and J. E. Flaherty. Parallel adaptive *hp*-refinement techniques for conservation laws. *Appl. Numer. Math.*, 20:367–386, 1996.
- [32] R. J. DiPerna. Measure-valued solutions to conservation laws. *Arc. Rat. Mech. Anal.*, 88:221–270, 1985.
- [33] C. A. Doswell. A kinematic analysis of frontogenesis associated with a nondivergent vortex. *J. Atmos. Sci.*, 41:1242–1248, 1984.
- [34] J. Duchon. Interpolation des fonctions de deux variables suivant le principe de la flexion des plaque minces. *R.A.I.R.O. Analyse Numeriques*, 10:5–12, 1976.
- [35] M. Dumbser and C. D. Munz. Arbitrary high order discontinuous Galerkin schemes. In M. G. S. Cordier, T. Goudon, and E. Sonnendrucker, editors, *IRMA series in Mathematical and Theoretical Physics*, pages 295–333. Zurich: EMS Publishing House, 2005.
- [36] G. E. Fasshauer. *Meshfree Approximation Methods in MATLAB*. World Scientific, 2007.
- [37] C. Franke. Solving partial differential equations by collocation using radial basis functions. *Appl. Math. Comput.*, 93:73–82, 1998.
- [38] O. Friedrich. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. *J. Comp. Phys.*, 144:194–212, 1998.
- [39] J. Fürst and T. Sonar. On meshless collocation approximations of conservation laws: positive schemes and dissipation models. *ZAMM*, 81:403–415, 2001.
- [40] E. H. Georgoulis, E. Hall, and P. Houston. Discontinuous Galerkin methods on *hp*-anisotropic meshes II: *a posteriori* error analysis and adaptivity. *Appl. Numer. Math.*, to appear.

-
- [41] S. K. Godunov. A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sbornik*, 47:357–393, 1959.
- [42] M. Golomb and H. F. Weinberger. Optimal approximation and error bounds. In R. E. Langer, editor, *On Numerical Approximation.*, pages 117–190. University of Wisconsin Press, 1959.
- [43] S. Gottlieb. On high order strong stability preserving Runge-Kutta and multi step time discretization. *J. Sci. Comput.*, 25:105–127, 2005.
- [44] S. Gottlieb and C. W. Shu. Total variation diminishing Runge-Kutta schemes. *Math. Comp.*, 67:73–85, 1998.
- [45] S. Gottlieb, C. W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization. *SIAM Review*, 43:89–112, 2001.
- [46] K. Guo, S. Hu, and X. Sun. Conditionally positive definite functions and Laplace-Stieltjes integrals. *J. Approx. Th.*, 74:249–265, 1993.
- [47] T. Gutzmer. Error estimates for reconstruction using thin plate spline interpolants. Technical Report 97-08, ETH, 1997.
- [48] T. Gutzmer and A. Iske. Detection of discontinuities in scattered data approximation. *Numer. Algor.*, 16:155–170, 1997.
- [49] A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comp. Phys.*, 49:357–393, 1983.
- [50] A. Harten. ENO schemes with subcell resolution. *J. Comp. Phys.*, 83:148–184, 1989.
- [51] A. Harten and S. R. Chakravarthy. Multidimensional ENO schemes for general geometries. Technical Report 91-76, ICASE, 1991.
- [52] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high order essentially non-oscillatory schemes, III. *J. Comp. Phys.*, 71:231–303, 1987.
- [53] A. Harten and S. Osher. Uniformly high order essentially non-oscillatory schemes, I. *SIAM J. Numer. Anal.*, 24:279–309, 1987.
- [54] C. Hirsch. *Numerical Computation of Internal and External Flows, Vol I.* John Wiley and sons, 1988.
- [55] E. V. Hólm. A fully two-dimensional, non-oscillatory advection scheme for momentum and scalar transport equations. *Mon. Wea. Rev.*, 123:536–552, 1995.

- [56] L. Hong, J. D. Baum, and R. Löhner. An efficient spatial adaption algorithm on 3-D unstructured meshes for the Euler equations. In *Proc. of the 5th International Symposium on CFD*, 1993.
- [57] P. Houston, B. Senior, and E. Süli. *hp*-discontinuous Galerkin finite element methods for hyperbolic problems: error analysis and adaptivity. *Inter. J. Numer. Meth. Fluids*, 40:153–169, 2002.
- [58] P. Houston and E. Süli. A note on the design of *hp*-finite element methods for elliptic partial differential equations. *Comput. Meth. Appl. Mech. Engrg.*, 194:229–243, 2005.
- [59] C. Hu and C. W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *J. Comp. Phys.*, 95:97–127, 1999.
- [60] M. E. Hubbard and N. Nikiforakis. A three-dimensional, adaptive, Godunov-type model for global atmospheric flows. *Mon. Wea. Rev.*, 131:1848–1864, 2003.
- [61] D. P. Hunt. *Meshfree radial basis function methods for advection-dominated diffusion problems*. PhD thesis, University of Leicester, 2005.
- [62] A. Iske. Reconstruction of functions from generalized Hermite-Birkhoff data. In Charles K. Chui and Larry L. Schumaker, editors, *Approximation Theory VIII*, pages 1–8. World Scientific Publishing Co., 1995.
- [63] A. Iske. On the approximation order and numerical stability of local Lagrange interpolation by polyharmonic splines. In W. Haussmann, K. Jetter, M. Reimer, and J. Stöckler, editors, *Modern Developments in Multivariate Approximation*, volume 145. Birkhäuser, Basel, 2003.
- [64] A. Iske. *Multiresolution Methods in Scattered Data Modelling*. Springer, 2004.
- [65] A. Iske and M. Käser. Conservative semi-Lagrangian advection on adaptive unstructured meshes. *Numer. Meth. Part. Diff. Eqns.*, 20:388–411, 2004.
- [66] A. Iske and M. Käser. Two-phase flow simulation by AMMoC, an Adaptive Mesh-free Method of Characteristics. *Comp. Mod. Eng. Sci.*, 7:133–148, 2005.
- [67] A. Iske and T. Sonar. On the structure of function spaces of optimal recovery of point functions for ENO schemes by radial basis functions. *Numer. Math.*, 74:177–201, 1996.
- [68] G. S. Jiang and C. W. Shu. Efficient implementation of weighted ENO schemes. *J. Comp. Phys.*, 126:202–228, 1996.

- [69] E. J. Kansa. Multiquadrics - a scattered data approximation scheme with applications to computational fluid dynamics - I. Surface approximations and partial derivative estimates. *Comput. Math. Appl.*, 19:127–145, 1990.
- [70] S. Karni, A. Kurganov, and G. Petrova. A smoothness indicator for adaptive algorithms for hyperbolic systems. *J. Comp. Phys.*, 178:323–341, 2002.
- [71] G. Karniadakis and S. Sherwin. *Spectral/hp element methods for CFD*. Oxford University Press, 2002.
- [72] M. Käser. *Adaptive methods for the numerical simulation of transport processes*. PhD thesis, Technical University Munich, 2003.
- [73] M. Käser and A. Iske. ADER schemes on adaptive triangular meshes for scalar conservation laws. *J. Comp. Phys.*, 205:486–508, 2005.
- [74] D. Kröner. *Numerical Schemes for Conservation Laws*. John Wiley and Sons, 1997.
- [75] A. Kurganov, G. Petrova, and B. Popov. Adaptive semi-discrete central-upwind schemes for nonconvex hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 29:2381–2401, 2007.
- [76] P. D. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure. Appl. Math.*, 13:217–237, 1960.
- [77] R. J. Leveque. High-resolution conservative algorithms for advection in incompressible flow. *SIAM J. Numer. Anal.*, 33:627–665, 1996.
- [78] R. J. Leveque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.
- [79] D. Levy, G. Puppo, and G. Russo. A third order central WENO scheme for 2D conservation laws. *Appl. Numer. Math.*, 33:415–421, 2000.
- [80] S. Li and J. Hyman. Adaptive mesh refinement for finite difference WENO schemes. Technical Report LA-UR-03-8927, Los Alamos National Laboratory, 1997.
- [81] X.D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comp. Phys.*, 115:200–212, 1994.
- [82] R. A. Lorentz. Collocation discretizations of the transport equation with radial basis functions. *Appl. Math. Comput.*, 145:97–116, 2003.

-
- [83] J. Meinguet. Multivariate interpolation at arbitrary points made simple. *Z. Angew. Math. Phys.*, 30:292–304, 1979.
- [84] A. Meister and J. Struckmeier. *Hyperbolic Partial Differential Equations: Theory, Numerics and Applications*. Springer, 2002.
- [85] C. A. Micchelli. Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constr. Approx.*, 2:11–22, 1986.
- [86] C. A. Micchelli and T. J. Rivlin. A survey of optimal recovery. In C. A. Micchelli and T. J. Rivlin, editors, *Optimal Estimation in Approximation Theory*, pages 1–54. Plenum, New York, 1977.
- [87] K. W. Morton and D. F. Mayers. *Numerical Solution of Partial Differential Equations*. Cambridge University Press, 1994.
- [88] F. J. Narcowich and J. D. Ward. Norm estimates for the inverses of a general class of scattered-data radial-function interpolation matrices. *J. Approx. Theory*, 69:84–109, 1992.
- [89] H. Nessyahu, T. Tassa, and E. Tadmor. The convergence of Godunov type schemes. *SIAM J. Numer. Anal.*, 31:1–16, 1994.
- [90] R. H. Nochetto, A. Schmidt, and C. Verdi. A posteriori error estimation and adaptivity for degenerate parabolic problems. *Math. Comp.*, 69:1–24, 2000.
- [91] J. O’Rourke. *Computational Geometry in C*. Cambridge University Press, 1993.
- [92] S. Osher. High resolution schemes and the entropy condition. *SIAM J. Numer. Anal.*, 21:955–984, 1984.
- [93] D. Peaceman. *Fundamentals of Numerical Reservoir Simulation*. Elsevier Scientific Publishing Co., 1977.
- [94] T. N. Phillips and A. J. Williams. Conservative semi-Lagrangian finite volume schemes. *Numer. Meth. Part. Diff. Eqns*, 17:403–425, 1999.
- [95] M. J. D. Powell. The theory of radial basis functions in 1990. In W. A. Light, editor, *Advances in numerical analysis II: wavelets, subdivision and radial basis functions*, pages 105–210. Clarendon Press, Oxford, 1992.
- [96] M. J. D. Powell. The uniform convergence of thin plate spline interpolation in two dimensions. *Numer. Math.*, 68:107–128, 1993.

-
- [97] J. X. Qiu and C. W. Shu. Finite difference type WENO schemes with Lax-Wendroff-type time discretizations. *SIAM J. Sci. Comp.*, 24:2185–2198, 2002.
- [98] S. J. Ruuth and J. Spiteri. Two barriers on strong stability-preserving time discretization methods. *J. Sci. Comput.*, 17:211–220, 2002.
- [99] S. A. Sarra. Adaptive radial basis function methods for time dependent partial differential equations. *Appl. Numer. Math.*, 54:79–94, 2005.
- [100] R. Schaback. Error estimates and condition numbers of radial basis function interpolation. *Adv. Comp. Math.*, 3:251–264, 1995.
- [101] R. Schaback. *Reconstruction of Multivariate Functions from Scattered Data*. Unpublished Text, 1997.
- [102] R. Schaback. Stability of radial basis function interpolants. In C. K. Chui, L. L. Schumaker, and J. Stoeckler, editors, *Approximation Theory X*, pages 1–8. Vanderbilt University Press, 2001.
- [103] I. J. Schoenberg. Metric spaces and completely monotone functions. *Ann. Math.*, 39:811–841, 1938.
- [104] C. Schwab. *p- and hp-Finite Element Methods*. Oxford Science Publications, 1998.
- [105] T. Schwartzkopff, C. D. Munz, and E. F. Toro. ADER: A high order approach for linear systems in 2D. *J. Sci. Comput.*, 17:231–240, 2002.
- [106] E. G. Sewell. *Automatic generation of triangulations for piecewise polynomial approximation*. PhD thesis, Purdue University, 1972.
- [107] J. Shi, C. Hu, and C. W. Shu. A technique of treating negative weights in WENO schemes. *J. Comp. Phys.*, 175:108–127, 2002.
- [108] C. W. Shu. Total variation diminishing time discretizations. *SIAM J. Sci. Stat. Comp.*, 9:1073–1084, 1988.
- [109] C. W. Shu. Numerical experiments on the accuracy of ENO and modified ENO schemes. *J. Sci. Comput.*, 5:127–149, 1990.
- [110] C. W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comp. Phys.*, 77:439–471, 1988.
- [111] C. W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes II. *J. Comp. Phys.*, 83:32–78, 1989.

- [112] J. Smit, M. van Sint Annaland, and J. A. M. Kuipers. Grid adaptation with WENO schemes for non-uniform grids to solve convection-dominated partial differential equations. *Chem. Eng. Sci.*, 60:2609–2619, 2005.
- [113] P. Smolarkiewicz. The multi-dimensional Crowley advection scheme. *Mon. Wea. Rev.*, 110:1968–1983, 1982.
- [114] J. Smoller. *Shock Waves and Reaction-Diffusion Equations*. Springer, New York, 1983.
- [115] T. Sonar. Optimal recovery using thin plate splines in finite volume methods for the numerical solution of hyperbolic conservation laws. *IMA J. Num. Anal.*, 16:549–581, 1996.
- [116] T. Sonar. On the construction of essentially non-oscillatory finite volume approximations to hyperbolic conservation laws on general triangulations : polynomial recovery, accuracy and stencil selection. *Comp. Meth. Appl. Engrg.*, 140:157–181, 1997.
- [117] T. Sonar. On families of pointwise optimal finite volume ENO approximations. *SIAM J. Numer. Anal.*, 35:2350–2369, 1998.
- [118] T. Sonar and E. Süli. A dual graph-norm indicator for finite volume approximations of the Euler equations. *Numer. Math.*, 78:619–658, 1998.
- [119] J. Spiteri and S. J. Ruuth. A new class of optimal high-order strong stability-preserving time discretization methods. *SIAM J. Numer. Anal.*, 40:469–491, 2002.
- [120] A. Staniforth, J. Cote, and J. Pudykiewicz. *Comments on "Smolarkiewicz's deformational flow"*. *Mon. Wea. Rev.*, 115:894–900, 1987.
- [121] B. Szabo and I. Babuška. *Finite element analysis*. John Wiley and sons, New York, 1991.
- [122] K. A. Tan, R. P. Morrison, and L. M. Leslie. A comparison of high-order explicit and non-oscillatory finite difference advection schemes for climate and weather models. *Met. Atmos. Phys.*, 89:251–267, 2005.
- [123] H. Z. Tang and T. Tang. Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws. *SIAM J. Numer. Anal.*, 41:487–515, 2003.
- [124] Z. Tang. A new approach to construct WENO schemes. In *Paper presented at Aerospace Science Meeting and Exhibit, Jan. 8-11, 2001*.

-
- [125] V. A. Titarev and E. F. Toro. ADER: arbitrary high order Godunov approach. *J. Sci. Comput.*, 17:609–618, 2002.
- [126] V. A. Titarev and E. F. Toro. Finite volume WENO schemes for three dimensional conservation laws. *J. Comp. Phys.*, 201:238–260, 2004.
- [127] V. A. Titarev and E. F. Toro. ADER schemes for three dimensional nonlinear hyperbolic systems. *J. Comp. Phys.*, 204:715–736, 2005.
- [128] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, 2nd edition.
- [129] E. F. Toro and R. C. Millington. The ADER approach to linear advection. In *ECCOMAS Computational Fluid Dynamics Conference*, 2001.
- [130] E. F. Toro, R. C. Millington, and L. A. M. Nejad. Towards very high order Godunov schemes. In E. F. Toro, editor, *Godunov Methods: Theory and Applications*, pages 907–940. New York: Kluwer/Plenum Academic Publishers, 2001.
- [131] E. F. Toro and V. A. Titarev. Solutions of the generalized Riemann problem for advection-reaction equations. In *Proceedings of the Royal Society of London*, pages 271–281, 2002.
- [132] E. F. Toro and V. A. Titarev. ADER schemes for scalar hyperbolic conservation laws with source terms in three space dimensions. *J. Comp. Phys.*, 202:196–215, 2004.
- [133] E. F. Toro and V. A. Titarev. Derivative Riemann solvers for systems of conservation laws and ADER methods. *J. Comp. Phys.*, 201:238–260, 2004.
- [134] B. van Leer. Towards the ultimate conservative difference scheme I. The quest for monotonicity. *Springer Lecture Notes Phys.*, 18:163–168, 1973.
- [135] B. van Leer. Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second order scheme. *J. Comp. Phys.*, 14:361–370, 1974.
- [136] B. van Leer. Towards the ultimate conservative difference scheme III. Upstream-centred finite difference schemes for ideal compressible flow. *J. Comp. Phys.*, 23:263–275, 1977.
- [137] B. van Leer. Towards the ultimate conservative difference scheme IV. A new approach to numerical convection. *J. Comp. Phys.*, 23:276–299, 1977.

-
- [138] B. van Leer. Towards the ultimate conservative difference scheme V. A second order sequel to Godunov's method. *J. Comp. Phys.*, 32:101–136, 1979.
- [139] S. Vukovic, N. Crnjacic-Zic, and L. Sopta. WENO schemes for balance laws with spatially varying flux. *J. Comp. Phys.*, 199:87–109, 2004.
- [140] R. Wang and R. J. Spiteri. Linear instability of the fifth-order WENO method. *SIAM J. Numer. Anal.*, 45:1871–1901, 2007.
- [141] H. Wendland. Piecewise polynomial, positive definite and compactly supported RBFs of minimal degree. *Adv. Comput. Math.*, 4:389–395, 1995.
- [142] H. Wendland. Meshless Galerkin methods using radial basis functions. *Math. Comp.*, 68:1521–1531, 1999.
- [143] H. Wendland. On the convergence of a general class of finite volume methods. *SIAM. J. Numer. Anal.*, 43:987–1002, 2005.
- [144] H. Wendland. *Scattered Data Approximation*. Cambridge University Press, 2005.
- [145] W. R. Wolf and J. L. F. Azevedo. High-order ENO and WENO schemes for unstructured grids. *Int. J. Numer. Meth. Fluids*, 55:917–943, 2007.
- [146] Z. Wu. Multivariate compactly supported positive definite radial functions. *Adv. Comput. Math.*, 4:283–292, 1995.
- [147] Z. Wu and R. Schaback. Local error estimates for radial basis function interpolation of scattered data. *IMA J. Numer. Anal.*, 13:13–27, 1993.
- [148] S. T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comp. Phys.*, 31:335–362, 1979.
- [149] Y. T. Zhang and C. W. Shu. High-order WENO schemes for Hamilton-Jacobi equations on triangular meshes. *J. Sci. Comput.*, 24:1005 – 1030, 2002.